NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

A SEAKEEPING STUDY ON THE AUTONOMOUS SUSTAINMENT CARGO CONTAINER DELIVERY SYSTEM

by

James W. Hedderly

March 2008

Thesis Advisor: Fotis Papoulias

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2008	**3. REPORT TYPE AND DATES COVERED** Mechanical Engineer's Thesis	
4. TITLE AND SUBTITLE: A Seakeeping Study on the Autonomous Sustainment Cargo Container Delivery System			**5. FUNDING NUMBERS**
6. AUTHOR(S) James W. Hedderly			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

An increasing emphasis has recently been placed with fighting non-national and irregular forces. Single entity attacks on shipping and transportation units have replaced attacks on fortified or established military positions. The supply chain from sea to shore has become the target of opportunity to disrupt the Global War on Terrorism and put further lives in danger. Autonomous containers will play an essential role to deliver logistical supplies to waterborne littoral vessels in order to maintain station and complete military operations all while eliminating the threat to human life as the containers will be programmed to deliver supplies to a specified local in a reasonable timetable; vessels such as Riverine Warfare patrol craft, Special Operations craft and Coast Guard search and rescue boats and their crews. The research to be conducted will focus on the seakeeping characteristics of an autonomous sustainment cargo container and the feasibility of its deployment. Established geometric data will be used along with changing loading characteristics and ride effects. The in depth analysis will be focused on the responses of the container in varying sea conditions and at varying loads to see if further refinement of the design or policies concerning loading and deployment may be required.

14. SUBJECT TERMS Autonomous, Cargo Container, Pitch, Slam, Seakeeping, Stability, Mast Submergence			**15. NUMBER OF PAGES** 77
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

A SEAKEEPING STUDY ON THE AUTONOMOUS SUSTAINMENT CARGO CONTAINER DELIVERY SYSTEM

James W. Hedderly
Lieutenant, United States Navy
B.S., United States Naval Academy, 1999

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
March 2008

Author: James W. Hedderly

Approved by: Fotis Papoulias
Thesis Advisor

Anthony J. Healey
Chairman, Department of Mechanical and Astronautical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

An increasing emphasis has recently been placed with fighting non-national and irregular forces. Single entity attacks on shipping and transportation units have replaced attacks on fortified or established military positions. The supply chain from sea to shore has become the target of opportunity to disrupt the Global War on Terrorism and put further lives in danger. Autonomous containers will play an essential role to deliver logistical supplies to waterborne littoral vessels in order to maintain station and complete military operations all while eliminating the threat to human life as the containers will be programmed to deliver supplies to a specified local in a reasonable timetable; vessels such as Riverine Warfare patrol craft, Special Operations craft and Coast Guard search and rescue boats and their crews. The research to be conducted will focus on the seakeeping characteristics of an autonomous sustainment cargo container and the feasibility of its deployment. Established geometric data will be used along with changing loading characteristics and ride effects. The in depth analysis will be focused on the responses of the container in varying sea conditions and at varying loads to see if further refinement of the design or policies concerning loading and deployment may be required.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. THE AUTONOMOUS SUSTAINMENT CONTAINER CONCEPT

A. BACKGROUND

Throughout the history of sea to shore engagements, supplying goods from units at sea to forces ashore has met with several shortcomings. Most notably, in the World Wars, Korea, and the recent conflicts in the Middle East, continuing to support troops ashore from forces afloat carries with it high risk, both to materiel and personnel. The current combat environment has placed further strain on this link of the supply chain as non-national forces look for easier and faster ways of harming the war effort than attacking fortified and/or heavily manned emplacements and convoys. A new method for delivering the necessary sustainment goods to forces ashore without endangering lives or providing easy targets of opportunity for these opposing forces needs be found.

B. HISTORY OF SEA TO SHORE SUSTAINMENT

1. Traditional Methods

To provide supplies to forces on shore, the method used throughout history required the supply ship to dock with a physical structure designed to move goods from ship to shore. As ships, especially cargo ships, grew in size, so did the required dock size. The dawn of the 20th century saw cargo ships that were too large to traditionally dock that moved supplies from anchor via smaller utility boats. This method still required a ship to berth at a physical dock to unload supplies. World War II saw the advent of supply sustainment without the physical dock. Army ships simply pulled up to the beach and offloaded supplies. This method of beach landing to resupply troops on the beach has remained the preferred mode since that time. As seen in Figure 1, U.S. troops deploy from an Army LCV on D-Day 1944 compared to U.S. and Royal troop landing in 2007. Although the commissioning dates of these two vessels are significantly different, the methodology remains primarily the same. The same is true about the delivery of parts and supplies from sea to shore.

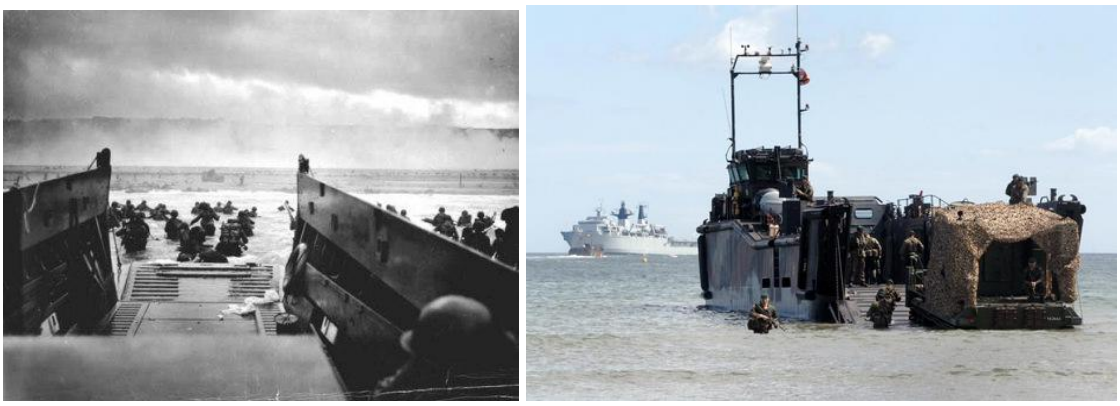


Figure 1. **U.S. Marine landing at Omaha Beach, 1944 (left) (From: 10) Joint U.S. and Royal Army troop landing 2007 (right) (From: 11)**

2. Non-traditional Techniques

There are points in history, where circumstances prevented the use of tradition supply delivery methods. Winds, waves, prevailing currents and even single individuals were used to circumvent the need for both berth and beach, or even for the ship to come in contact with land at all. "…the Tokyo Express was [having difficulty delivering supplies]…so the measures now being taken were desperate ones. Destroyers tried to supply these Imperial troops by making high-speed runs to Guadalcanal and dropping off strings of lashed-together drums into which supplies had been sealed. Barges from the island then were to tow these drums ashore. The troops managed to retrieve only about 30 per cent of these supplies that Tanaka's destroyers cast upon the water. Tanaka's destroyers raced down The Slot on 3 December and dropped strings of 1,500 drums. But the island troops managed to haul in only about 300 of these from the waters off Tassafaronga and Segilau. On 11 December, Tanaka himself led one in his speedy *Teruzuki…* and dropped some 1,200 drums of supplies off Cape Esperance and then headed north again. Only some of the drums were recovered by the troops ashore…"[5] The plight of the Japanese troops continued until the islands were finally conquered by U.S. troops. Further stories of the Tokyo Express include using divers to clandestinely recover supplies floated by Japanese destroyers and midnight barge runs through U.S. patrolled waters. When the traditional ways begin to fail, it is essential to make every effort to solve the problem with something new.

3. Unmanned Ships in History

There have been a few instances of unmanned and semi-autonomous vessels taking part in important naval operations where manned vessels simply would not have been as capable. One of the most famous uses of unmanned craft occurred during the battle between the Spanish Armada and the Queen's Fleet off of Calais. Sir Francis Drake and John Hawkins hoped to pull the Armada out of safe harbor using the then famous hellburners of Antwerp, ships that had been made into tremendous bombs, as the tool in which to bring the Spanish fleet into combat. "Drake offered a ship of his own, the *Thomas* of Plymouth, of two hundred tons, and Hawkins one of his; and with the mounting enthusiasm six more were recruited...the captains scattered to the work of getting them ready and stuffing them with everything available to make a blaze."[3, p271] "But not only were spars and sails and rigging left, since the ships would bear down on the anchorage…under full sail, but all the ships' guns were left double-shotted to go off when the fire made them hot enough, to add to the terror if not the destruction of the enemy."[3, p271] "…the Spanish anchorage could see plainly eight tall ships with all sails set and lines of fire beginning to run up their rigging, driving straight towards them with wind and tide."[3, p277] "…the line was driving along so fast, pushed by a strong wind, a spring tide and the set of the Channel current all working together…"[3, p277] "…the double shotted guns heated now white hot, began to go off, spraying their shot at random over the water and by the force of their recoil sending up a fountain of sparks to be blown among the boats…the six remaining ships swept past and bore down on the anchored fleet, the sound of their exploding guns heard above the roar of the flames and the fountains of sparks shooting skyward."[3, p277] " …most captains simply cut their cables and ran before the wind, scattering, some here, some there, as if they were as much afraid of one another as they were of the fire-ships. The strong set of the current and the rising gale swept the whole disorderly mob out through the straits and on toward the sands of the Flemish coast."[3, p278] The autonomous fire-ships did little physical damage to the Spanish fleet, they were after all not really the hellburners the Spanish had feared, but when dawn approached the Armada was in complete disarray. Unable to

reform, they were harried for days by the Queen's fleet until able to flee northward. This marked the end of the Spanish Order and set about the beginning of the English domination of the sea.

4. The Emphasis on Unmanned Craft

With the development of new concepts in military operations and reductions in manpower of our military forces, the promotion of autonomous systems has been pushed to the forefront. Recent improvements in wireless and satellite communications, sensor and data relay systems and remote operability has made unmanned vehicles an available option for easing the manpower burden felt by much of the military. An autonomous vehicle is a self-piloted vehicle that does not require an operator to navigate and accomplish its tasks. As the necessity to protect human life grows, the need for autonomous vehicles will grow as well. The U.S. Air force has proven, in combat, the range of uses for unmanned vehicles. The Air force Predator has run countless sorties in Afghanistan and Iraq with missions ranging from simple surveillance to offensive surgical strikes. The pilot does all this from a remote operating station well away from the threat region. Building off of the advances and successes of the Predator, Global Hawk will fulfill mission roles from long range surveillance and tracking to bombing strikes against hardened, buried and bunkered targets. Further UAV systems are in the works including the Boeing stealth UAV, the MQ-8B Fire Scout helicopter UAV, and the autonomous programmable Silver Fox and Scan Eagle UAVs developed at the Naval Postgraduate School. These advances have spread to surface and subsurface craft as well. Current systems in development include the BAE system Protector USV for Coast Guard applications, and the autonomous Sea Fox USV and Remus UUV.

C. THE DIFFICULTIES OF THE SEA-SHORE INTERFACE

The sea shore interface is an area rife with hazards. Shallow and shoal waters are home to shipping dangers such as reefs, mine hazard areas, capsizing surf conditions, and underwater rock outcroppings. The shoals are further threatened during combat as the threat level increases with enemy activities. For the Marines at Guadalcanal the problem was service support facilities. "Without port facilities, supplies reached the troops only

after a series of time-consuming and labor-intensive equipment transfers. Supplies were first unloaded from Navy ships offshore into lighters for the trip to the beach. There American service support personnel transferred the tonnage to trucks that hauled it inland to several dumps on roads under construction. From the dumps supplies had to be hand carried, by both Americans and native laborers… As the fighting moved further inland the distance between dumps and front line lengthened and road building could not progress as fast as assault units advanced… American troops temporarily solved the distribution problem by using the many streams and rivers on the island. Loading supplies into small boats…Americans pushed the craft through the water as close to the tactical units as possible."[4, p26-27] The 1st Marine Division faced problems from both the shoals and the enemy as they tried to get supplies onto the beach at Inchon. "At Inchon, the lack of adequate port facilities made it essential that the naval logistics forces rapidly move supplies 'over the beach.'"[6, p40] While the UDT cleared obstacles and the Navy Seabees built a large pontoon dock and reconstructed port facilities, eight LSTs beached themselves, under fire, and began the task of disgorging tons of supplies and munitions. "After the evening landing, some of [Captain Watson] Singer's sailors and [General Henry] Crowe's marines had to unload, organize and distribute the supplies from the eight LSTs temporarily immobilized on Red Beach…"[6, p40] The precarious condition of these beached vessels can be seen in Figure 2, but the surf and sand wasn't the only threat to these vessels or their crews. "Enemy snipers in Inchon shot at Crowe's leathernecks and Singer's sailors, outlined as they were in the glare of the floodlights installed by the Seabees. Nonetheless, the men continued to toil throughout the long night to accomplish their vital mission. All of the LST captains withdrew their emptied ships on the morning's high tide."[6, p40]

Figure 2. **Four "high-and –dry" LSTs disgorge supplies onto Red Beach on or about 16 September (1950) (From: 6)**

D. THE ASCC MISSION

1. Purpose

Waterborne logistics delivery systems such as the mechanized Landing Craft Utility (LCU), Landing Craft Air Cushion (LCAC), and High Speed Vessels (HSV) have been used for logistical delivery for many decades. All three of those types of delivery vehicles are very capable at providing logistic support with speed and efficiency; however, they all provide the opportunity for the loss of human life. The ASCC is designed to complete the same missions with equal or better speed, cost and efficiency, to effectively remove the need to task those particular vessels with logistic support missions. The overall main purpose of the ASCC is to replace the need for the logistical interface of sea basing. With the reduction of sea basing comes with it the reduction of bulk cargo constraints and cargo breakouts, manpower and the addition in time savings. The mission application has immense appeal to Joint Logistics over the Shore (JLOTS). The ASCC goes further; to reduce the threat to crew and service support personnel aboard both the sea based logistics ship and the delivery vessel itself. The reduction in manpower requirements, enhancements of personnel safety, and reduction of local infrastructure requirements are its greatest attractors.

2. Mission Profile

Figure 3 shows a hypothetical coastline where troops positioned ashore are expecting logistic support. The opposing forces have mined the coastline and set up surface to surface missile sights overlooking the bay. Under conventional sea-shore support missions, the logistics craft would have to close to within a few miles of shore to offload the supply carriers. These supply carriers, LSVs, LCACs, etc., then must navigate through the mine fields as seen in Figure 4. This scenario would put a significant number of lives at risk and could easily lead to the loss of equipment, vessels, and personnel. Using the ASCC, the logistics vessel remains well outside of weapons range while offloading the ASCCs. The container would then travel along a defined route until it reaches its designated offload area. Once onshore, the ASCC could be unloaded manually, loaded on to cargo trucks for further transport, or reloaded with retrograde to return to the logistic ship off shore. Figure 5 shows the ASCCs route bypassing the mine field and supplying the troops on the beach with needed supplies.

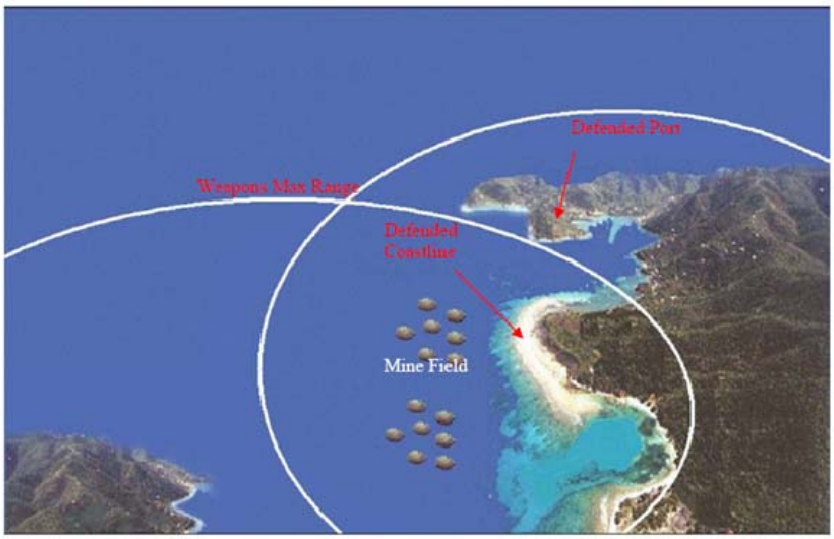


Figure 3. **The ASCC Operating Environment (From: 14)**

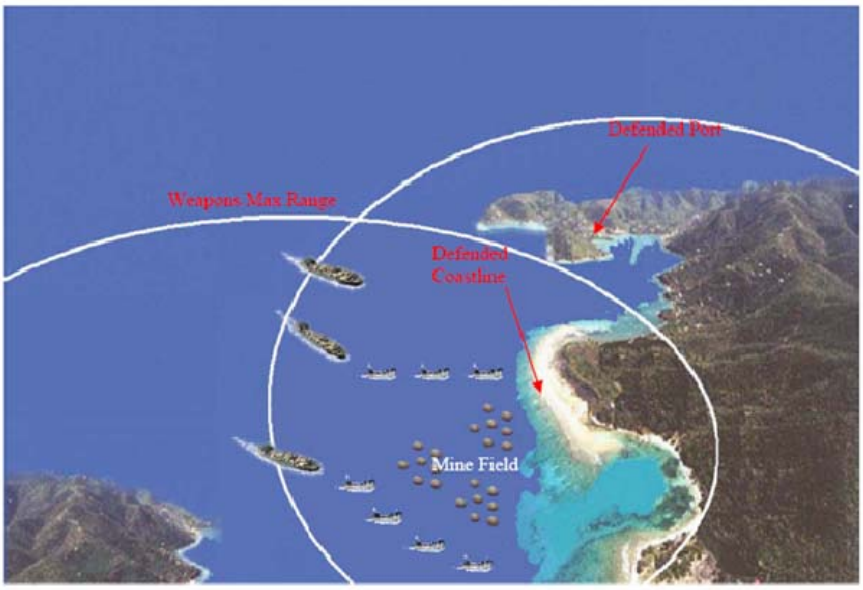


Figure 4. **The Threat to Personnel Using Traditional Methods (From: 14)**

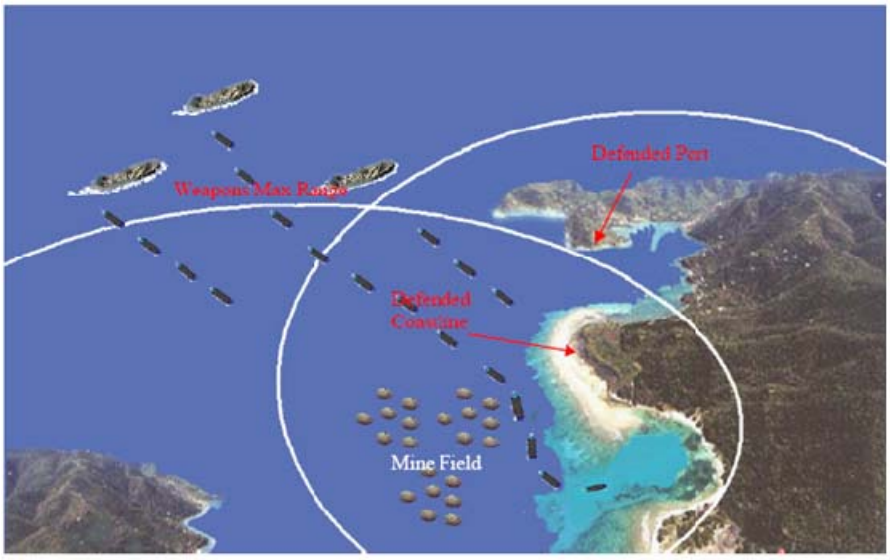


Figure 5. **The ASCC Delivery System (From: 14)**

3. The AEPCO Design

Advanced Engineering & Planning Corporation, Inc. (AEPCO) has proposed a Autonomous Sustainment Cargo Container that would solve the aforementioned problems. The ASCC aims to reduce manpower requirements for various littoral vessels as well as enhance personal safety by transporting cargo containers from the logistics

ship to shore using an autonomous system attached directly to the container. The typical ISO container can be seen in Figure 6 and comes in a 20 foot and 40 foot length. Initial designs have used to 20 foot ISO container to produce the vessel in Figure 7. The concept has a transit speed between 15 and 20 knots and enough range to transport goods from the logistics ship offshore, follow a programmed route past hazards, and arrive for pick up by troops ashore. AEPCO is currently involved in concept exploration in order to answer the fuel/power and speed/range questions.

Figure 6. **Typical ISO Container (From: 12)**

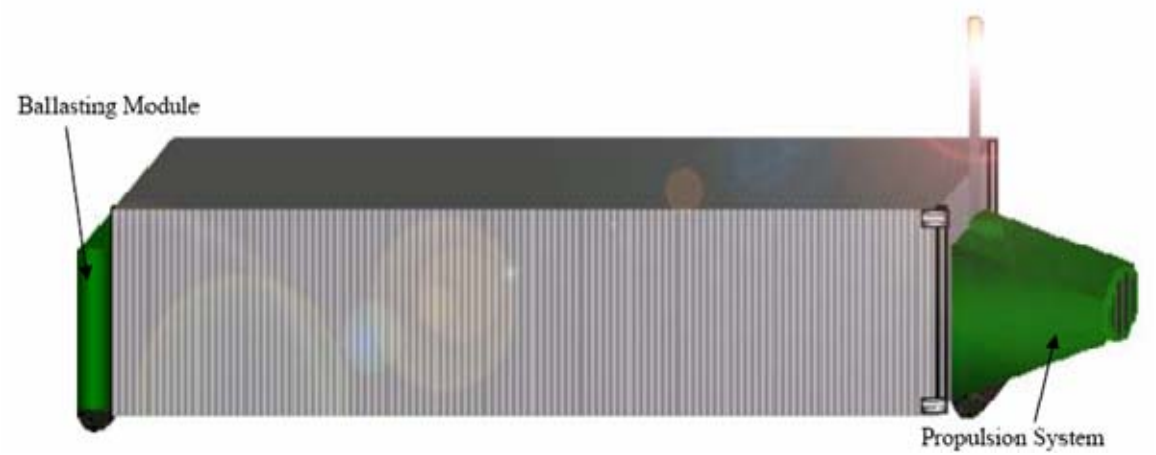


Figure 7. **AEPCO Proposed ASCC system (From: 13)**

E. THE PURPOSE OF THIS ANALYSIS

The purpose of this analysis is to perform a proof of concept on the AEPCO ASCC design with an emphasis on seakeeping. The ASCC parameters will be input into a computer simulation/calculation program and the results will be compared with

indicated ship performance parameters. Based on the results, areas of concern or design shortfalls will be further scrutinized with an eye toward making design improvement suggestions. The simulated model will test the vessel in the areas of pitch, roll, vertical acceleration, lateral acceleration, mast submergence, propeller racing due to broaching, and bow slamming. These results will then be compared to probable operational areas to determine the operational limits to the particular operating environment. This will then allow for the calculation and display of the overall operational capability of this vessel. It is not the intention of this study to conduct a detailed structural analysis of the ASCC or create a finite element model of the seas and their interaction with the vessel structure.

II. SEAKEEPING ANALYSIS

A. INITIAL SETUP

1. The Proposed Layout

The AEPCO ASCC is designed to use a standard 20 foot ISO container. It focuses on the 80% loading condition. This is both the most common and most economical way to deliver goods. This study will also use the 80% as the basis for analysis but will also add the 50, 60, and 70% loading conditions for comparison. Dimensions for the 20 foot ISO at 80% load can be seen in Table 1. This data was used to build the spreadsheet seen in Table 2. The ASCC KG calculation table is designed to solve for the various loading conditions based on a high, mid-box, or low loaded center of gravity. The distance from the keel to the center of gravity is called KG. The most common loading condition is the mid-box load where items inside the container are uniformly distributed and the center of gravity remains roughly in the center of the load. For ease of calculations, KG values were rounded to the nearest 10 for input into the analysis program. For the engine intake, a snorkel system has been designed to also act as a mast head to comply with Coast Guard Regulations. The mast head height is determined by the distance from which the mast head light can be seen. Coast Guard regulations require that partially submerged, unmanned vessels have a range of visibility of three miles.[8] This range converts to a required minimum mast head height of one meter. The one meter tall snorkel height is used for determining the mast submergence event occurrences.

Table 1. **Dimensions of a 20 ft ISO Container at 80% Load (From: 13)**

20' ISO Under Loading Condition	
% Loaded	80
▲ [lbs]	42316
Length (L) [ft]	19.875
Beam (B) [ft]	8
Height (H) [ft]	8
▲[LT]	18.891
▼[ft^3]	661.188
Draft (T) [ft]	4.158
Length-to-Beam Ratio (L/B)	2.48
Beam-to-Draft Ratio (B/T)	1.92
L' [ft]	35.875
L'/B	4.48

Table 2. **ASCC KG calculation**

		Percent Load										
		0	5	10	20	30	40	50	60	70	80	High Load
	weight	8002.78	10147.36	12291.93	16581.09	20870.24	25159.39	29448.55	33737.70	38026.85	42316.00	Midbox Load
	weight (LT)	3.57	4.53	5.49	7.40	9.32	11.23	13.15	15.06	16.98	18.89	Low Load
	V	125.04	158.55	192.06	259.08	326.10	393.12	460.13	527.15	594.17	661.19	
	T	0.79	1.00	1.21	1.63	2.05	2.47	2.89	3.32	3.74	4.16	
	KB	0.39	0.50	0.60	0.81	1.03	1.24	1.45	1.66	1.87	2.08	
	BM	54.25	42.79	35.32	26.19	20.80	17.26	14.74	12.87	11.42	10.26	
KG (est)	GM											
4.25		50.40	39.04	31.68	22.75	17.58	14.24	11.94	10.28	9.04	8.09	
4		50.65	39.29	31.93	23.00	17.83	14.49	12.19	10.53	9.29	8.34	
3.75		50.90	39.54	32.18	23.25	18.08	14.74	12.44	10.78	9.54	8.59	
3.5		51.15	39.79	32.43	23.50	18.33	14.99	12.69	11.03	9.79	8.84	
3.25		51.40	40.04	32.68	23.75	18.58	15.24	12.94	11.28	10.04	9.09	
3		51.65	40.29	32.93	24.00	18.83	15.49	13.19	11.53	10.29	9.34	
2.75		51.90	40.54	33.18	24.25	19.08	15.74	13.44	11.78	10.54	9.59	
2.5		52.15	40.79	33.43	24.50	19.33	15.99	13.69	12.03	10.79	9.84	
2.25		52.40	41.04	33.68	24.75	19.58	16.24	13.94	12.28	11.04	10.09	
2		52.65	41.29	33.93	25.00	19.83	16.49	14.19	12.53	11.29	10.34	
1.75		52.90	41.54	34.18	25.25	20.08	16.74	14.44	12.78	11.54	10.59	
1.5		53.15	41.79	34.43	25.50	20.33	16.99	14.69	13.03	11.79	10.84	
1.25		53.40	42.04	34.68	25.75	20.58	17.24	14.94	13.28	12.04	11.09	
1		53.65	42.29	34.93	26.00	20.83	17.49	15.19	13.53	12.29	11.34	

Metric		Percent Load									
		0	5	10	20	30	40	50	60	70	80
	weight	8002.78	10147.36	12291.93	16581.09	20870.24	25159.39	29448.55	33737.70	38026.85	42316.00
	weight (LT)	3.57	4.53	5.49	7.40	9.32	11.23	13.15	15.06	16.98	18.89
	weight (kg)	3630.06	4602.84	5575.62	7521.18	9466.74	11412.30	13357.86	15303.42	17248.98	19194.54
	V (m^3)	3.54	4.49	5.44	7.33	9.23	11.13	13.02	14.92	16.81	18.71
	T (m)	0.24	0.30	0.37	0.50	0.62	0.75	0.88	1.01	1.14	1.27
	KB	0.12	0.15	0.18	0.25	0.31	0.38	0.44	0.50	0.57	0.63
	BM	5.04	3.98	3.28	2.43	1.93	1.60	1.37	1.20	1.06	0.95
KG (est)	GM										
1.2954		53.35	41.99	34.63	25.70	20.53	17.20	14.90	13.23	11.99	11.04
1.2192		53.43	42.07	34.71	25.78	20.61	17.27	14.97	13.31	12.07	11.12
1.143		53.50	42.14	34.78	25.86	20.69	17.35	15.05	13.38	12.14	11.20
1.0668		53.58	42.22	34.86	25.93	20.76	17.43	15.12	13.46	12.22	11.27
0.9906		53.66	42.30	34.94	26.01	20.84	17.50	15.20	13.54	12.30	11.35
0.9144		53.73	42.37	35.01	26.09	20.91	17.58	15.28	13.61	12.37	11.43
0.8382		53.81	42.45	35.09	26.16	20.99	17.66	15.35	13.69	12.45	11.50
0.762		53.88	42.52	35.16	26.24	21.07	17.73	15.43	13.76	12.52	11.58
0.6858		53.96	42.60	35.24	26.31	21.14	17.81	15.50	13.84	12.60	11.65
0.6096		54.04	42.68	35.32	26.39	21.22	17.88	15.58	13.92	12.68	11.73
0.5334		54.11	42.75	35.39	26.47	21.30	17.96	15.66	13.99	12.75	11.81
0.4572		54.19	42.83	35.47	26.54	21.37	18.04	15.73	14.07	12.83	11.88
0.381		54.27	42.90	35.54	26.62	21.45	18.11	15.81	14.15	12.91	11.96
0.3048		54.34	42.98	35.62	26.69	21.52	18.19	15.89	14.22	12.98	12.03

2. Modeling Waves and Sea States

Vessel motions in regular waves were found using the strip theory of Salveson, Tuck, and Faltinsen as modified by Beck to include the surge degree of freedom.[3] The linear simultaneous equation that must be solved is $\sum_{k=1,6}^{j=1-6} (\omega_e^2 (M_{jk} + A_{jk}) + i\omega_e B_{jk} + C_{jk})\varsigma_k = F_j^I + F_j^D$ where ζ1-6 are surge, sway, heave, roll, pitch, and yaw respectively; M is the mass matrix, A is the added mass matrix, B is the damping matrix, C is the hydrostatic restoring force matrix, F^I is the Froude Krylovexciting force in the j^{th} mode of motion, and F^D is the diffraction exciting force in

the j[th] mode of motion. For a port and starboard symmetric ship the mass matrix is given as

$$M_{jk} = \begin{vmatrix} M & 0 & 0 & 0 & M\overline{z}_c & 0 \\ 0 & M & 0 & -M\overline{z}_c & 0 & M\overline{x}_c \\ 0 & 0 & M & 0 & -M\overline{x}_c & 0 \\ 0 & -M\overline{z}_c & 0 & I_{44} & 0 & -I_{46} \\ M\overline{z}_c & 0 & -M\overline{x}_c & 0 & I_{55} & 0 \\ 0 & M\overline{x}_c & 0 & -I_{46} & 0 & I_{66} \end{vmatrix}$$

where M is the vessels total mass, x, y, and z are the centers of mass in the longitudinal, transverse and vertical directions, and I is the mass moment of inertia about the subscripted axis. In long crested random seas the response spectrum is determined by the linear relationship $S_o+(\omega,\beta) = |H(\omega,\beta)|^2 S_I+(\omega)$ where S_I+(ω) is the one-sided incident wave spectrum, H(ω,β) is the response amplitude operator for waves propagating in direction β, and S_o+(ω,β) is the one-sided response spectrum due to waves traveling in direction β. A "cosine-squared" spreading function is used such that the area under the spectrum is the mean square value of the process. Several different wave spectra were used by the modeling program. The Pierson-Moskowitz spectrum as $S_I+(\omega) = 0.0081g^2/\omega^5 e^{(-0.32(g/H_s\omega^2)^2)}$ where H_s is the significant wave height, ISSC spectrum as $S_I+(\omega) = 0.11H_s^2\omega_m^4/\omega^5 e^{(-0.44(\omega_m/\omega)^4)}$ where $\omega_m = 2\pi/T_m$ and $T_m =$ mean period, ITTC spectrum based on zero crossings as $S_I+(\omega) = (H_s^2/4\pi)(\omega_m^4/\omega^5)e^{(-(1/\pi)(\omega_z/\omega)^4)}$ where $\omega_z = 2\pi/T_z$ and $T_z =$ zero crossing period, ITTC spectrum based on modal (or peak) period as $S_I+(\omega) =).3125H_s^2\omega_p^4/\omega^5)e^{(-1.25(\omega_p/\omega)^4)}$ where $\omega_p = 2\pi/T_p$ and $T_p =$ period of spectral peak, the Ochi 6-parameter spectrum as

$$S_I+(\omega) = 1/4\sum_{j=1,2}[\{0.25(4\lambda_j+1)\omega_{mj}^4\}\lambda_j/\Gamma(\lambda_j)]\times[H_{sj}^2/\omega(4\lambda_j+1)]\times e^{\{-0.25(4\lambda_j+)(\omega_{mj}/\omega)^4\}}$$

where $\omega_{mj} =$ modal frequency of jth hump, $\lambda_j =$ shape parameter for jth hump, and

$H_{sj} =$ significant wave height of jth hump, and JONSWAP spectrum as

$$S_I + (\omega) = \alpha g^2 \omega^{-5} e^{(-1.25(\omega_m/\omega)^4)} \times \gamma e^{(-(\omega-\omega_m)^2/(2\sigma^2\omega_m^2))}$$ where

$X = gx/U_{10}^2$

$U_{10} =$ wind speed at 10 meters

$x =$ fetch in meters

$\alpha = 0.076X^{-0.22}$

$\omega_m = 2\pi 3.5X^{-0.33}(g/U_{10})$

$\gamma = 3.3$

$\sigma = 0.07$ for $\omega \le \omega_m$

$\sigma = 0.09$ for $\omega > \omega_m$

Further refinement of the wave spectrum is done using the defined sea state significant wave height and modal wave period found in Table 3. The region specific percent probability values are used for later operability calculations.

Table 3. **Sea State Definition and Example Region Probabilities**

Sea State	Significant Wave Height (m)	Modal Wave Period (sec)	Percent Probability (Region specific)
1	0.1	6.3	1
2	0.4	6.5	6
3	1.1	7.5	15
4	2.4	8.8	32
5	3.5	9.7	21
6	5.5	12.4	15
7	8.0	15.0	8
8	12.0	16.5	2

3. Standards of Measurement for Naval Vessels

The ASCC parametric limits were based on a combination of Naval standards shown in Figure 8, operational requirements, and mechanical limitations. Due to the unique nature of the ASCC, its lack of crew, radar, and sensitive electronics, motion extremes such as pitch, roll, slam, vertical acceleration and acceleration, and lateral velocity do not need to remain as fixed as the values listed in Figure 8. Pitch, roll, vertical and lateral accelerations, slam are generally associated with crew comfort. These motions contribute to seasickness. For the ASCC, seasickness in the crew is not an issue and the parametrics can be freely modified to reflect the unique nature of this vessel.

These malleable limits were adjusted as follows: pitch limits were increased to 10 degrees, roll went to 15 degrees, slam became 1.0 g's, vertical acceleration was increased to 1.0 g's, vertical velocity went to 3.0 m/s, and lateral accelerations became 0.5 g's. For propeller racing events Figure 8 shows a limit of 25/100 which when calculated for a design trip of 40 NM leads to a value of 350 racing events per hour. A new category, mast submergence events, was added due to the damage that could occur should the engine intake pull a slug of water 10 cm or greater in length. For this study, the limit was set to the most conservative value, one event per trip.

Table 20—Prescribed Values for the Twelve Performance Criteria of Table 19

Criterion No.	Column No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14
	Ship type	Naval mono-hull		3350 t. SWATH	Planing craft	Hydro-foil	2000–3000 t SES	Naval mono-hull		Naval mono-hull		Commerical Monohulls			
												Bulk Carrier	Gen. Cargo L.	Trawler	Cross-chan.
	Function	Point-to-point transit						Helicopter operation		At sea replen.	Sonar search	Transit			
	Source	Olson (1977)	Comstock et al. (1980)	Olson. (1977)	Allen et al (1978)	Stark (1977)	Mandel (1979)	Olson (1977)	Comstock et al. (1980)	Olson (1977)	Olson (1977)	Aertssen (1968, 1972)			
1	Rms Roll Angle, deg.	9.6	4.	9.6	—	1.25	1.5	3.2	2.5	—	9.6	—	—	—	—
2	Rms Pitch Angle, deg.	—	1.5	—	—	1.5	1.5	—	1.5	2.4	—	—	—	—	—
3	Rms Vert. Disp., m.	—	—	—	—	—	—	1.26	—	—	—	—	—	—	—
4	Rms Vert. Accel., g	—	0.2	—	—	0.11	0.1	—	0.2	—	—	0.5	0.9	1.4	1.0
5	Rms Lat. Accel., g	—	0.1	—	—	0.06	0.1	—	0.1	—	—	—	—	—	—
6	Mot. Sickness Incidence	20% in 2 hrs	—	20% in 2 hrs.	—	10% in 4 hrs.	10% in 2 hrs.	—	—	—	20% in 2 hrs.	—	—	—	—
7	Slam Accel., g.	0.2	—	—	4*	0.5*	0.6*	0.2	—	0.2	0.2	0.2	0.2	0.9	0.4
8	Slam Freq.	1/100	2%	#	—	—	—	1/100	2%	1/100	1/100	1/100	1/100	1/100	1/100
9	Sonar Dome Emerg. Freq.	—	—	—	—	—	—	—	—	—	1%	—	—	—	—
10	Dk. Wetness Freq.	5%	3%	—	—	—	—	5%	8%	5%	8%	1/100	1/100	—	—
11	Prop. Emerg. Freq.	—	—	#	—	—	—	—	—	—	—	95/100	85/100	—	—
12	Rms Rel. V. Velocity, m/s	—	—	—	—	—	—	1.83	1.0	—	—	—	—	—	—

Notes: All values of response (angles, displacements, accelerations) are rms (root mean square) peak-to-mean (single amplitude) values unless otherwise noted.
* Extreme peak-to-mean value.
# See text, Section 7.6

Seakeeping Performance Criteria—Table 2

Figure 8. **12 Prescribed Values for Naval Vessel Performance (From: 1)**

B. DESIGN ANALYSIS

1. SHIPMO

The program used to model the waves and wave responses is called SHIPMO.BM. The program predicts ship motions in six degrees of freedom (6-DOF) and five components of the shear and bending moment distributions. It is based on the strip theory approach of Salvesen, Tuck, and Faltinsen which was extended by Beck to include the surge degree of freedom. The program computes both regular and irregular waves and uses a cosine-squared spreading function for short crested seas. The program is based on the slender body theory, which assumes a long slender vessel with fine ends.

The version used for this analysis has been updated to include barge type forms with blunt ends which take into account the end effects. This is of particular note for the box shaped ASCC. 9,196 runs were made using loads from 50 to 80 percent in 10 percent intervals, sea states from 1 to 8, and design speeds from 0 to 20 knots.

a. Data Input

Vessel specific data and run dynamics are input in a Fortran 77 format. The ships length, breadth, and design draft in meters are entered in at line 3. The vessels hull form is entered as a series of discrete points as seen in Figure 9. KG in meters measured from the waterline is entered as a difference with draft in line 8. The range of speeds in meters per second is entered into line 9. The range of headings in degrees for seas is in line 11. A sample input file can be found in Appendix B. Every change in draft, KG, and sea state becomes a new input file, which leads to a large number of required runs to cover all likely combinations.

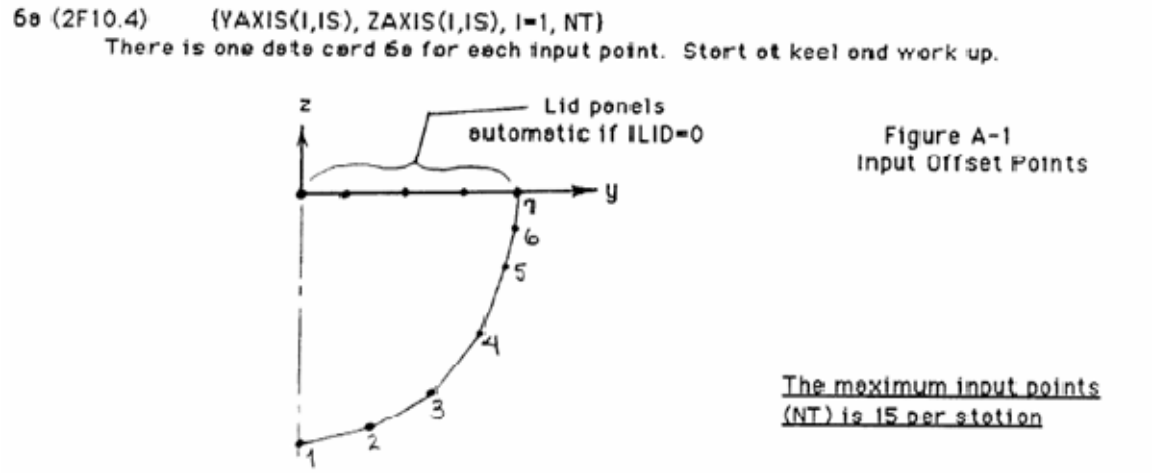


Figure 9. **Input Points Format for SHIPMO (From: 2)**

b. Data Extraction

The process through which the SHIPMO program calculates the modeling results can be seen in the Computer Program Flow Chart in Figure 10 and 11. Each run in SHIPMO is output as a *.txt table and *.emf graphics file. These outputs are imported and collated through a previously created Matlab code shown in Appendix A. This code pulls together the sea state and load results and builds a graphic representation as

compared to the input ship limits as determined by ship standards in part A3 and mission requirements. The results come out as a polar plot with ship heading toward the polar left, so waves striking the ships bow will be seen coming from the 270 degree point on the polar plot. Speed rings are measured from the center being zero, the ¼ ring at 5 knots, the ½ ring at 10 knots, the ¾ ring at 15 knots, and the outer ring at the max 20knots. Measured results that exceed the design parameters are noted in red, the various other shades note degrees to which the vessel is approaching those limits.

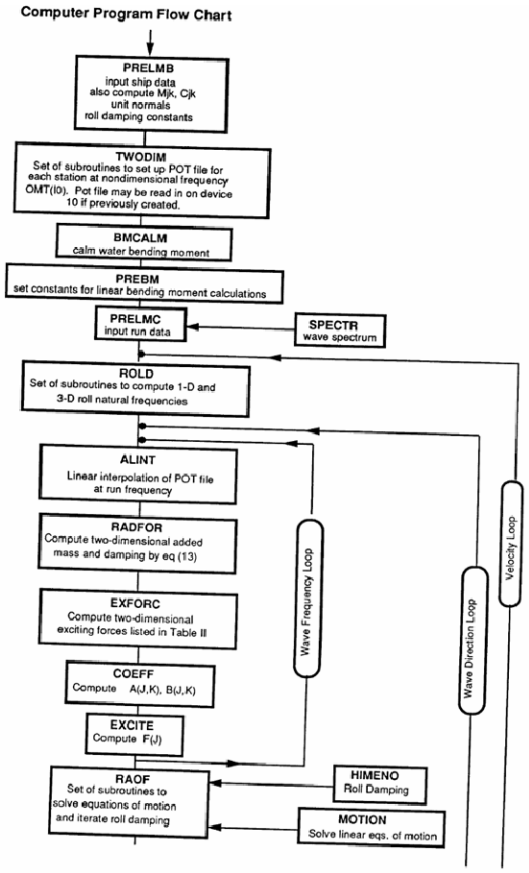


Figure 10. **The SHIPMO Computer Program Flow Chart (From: 2)**

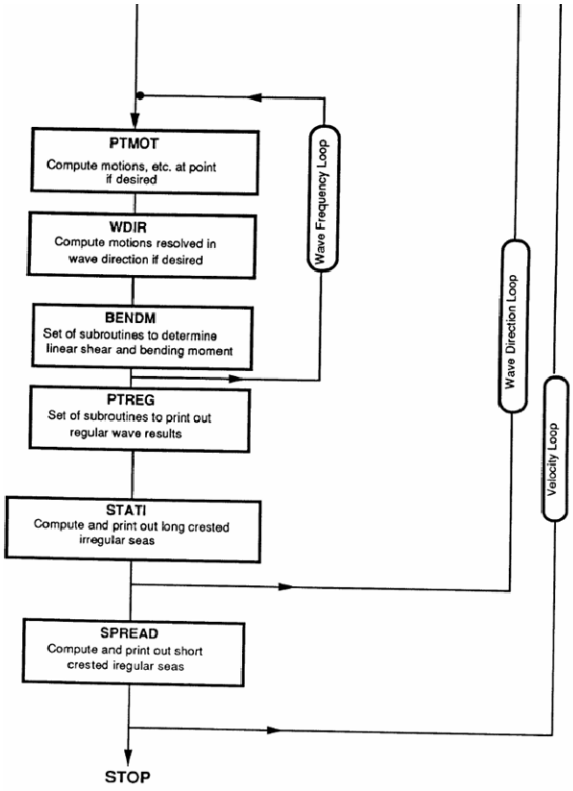


Figure 11. **The SHIPMO Computer Program Flow Chart cont. (From: 2)**

C. RESULTS

1. Initial Findings

Preliminary review of the outputs showed a startling number of red areas in the 15 to 20 knot speed range in head and fore quartering seas. Selected results can be found in Appendix B. The output results for pitch, roll, slam, mast submergence, propeller racing, and lateral acceleration were review for potential problems in the design. The results were also put together to help determine what categories were of more particular importance then others. Many of the results are based on the principles of ship motion involving crew comfort and operational windows for crew laded vessels. Seasick causing motions like lateral and vertical accelerations can be completely ignored due to the lack of crew on the ASCC. Motions which cause seasickness but also have the potential to do damage, like slamming, can also be shown to be outside the scope of this study as the structural strength or placement of delicate components are not included. Propeller racing events although significant are less of a concern as the location of the propeller was not known and the data was estimated for a propeller 20 and 30 cm from the keel.

Additionally, Caterpillar diesels are extremely hardened against this type of marine environment abuse. After careful review, pitch, roll and mast submergence showed themselves as the most particularly significant results obtained. Mast submergence is especially important due to the possible damage that water intrusion can do to the Caterpillar diesel [15]. Both pitch and roll motions can be shown to be the root cause to both propeller racing and mast submergence events and are the driving factors to the limits to the ASCC's operational capabilities.

2. Results of Particular Importance

a. Mast Submergence

Based on documentation provided by Caterpillar Marine, the proposed supplier of the ASCCs prime mover, water intrusion of any amount into the intake would result in serious damage [15]. This damage is not relegated to simply the intake but could damage the manifold, turbo, exhaust and the block itself. Additionally, a shortage of combustion air due to a water slug in the intake tube would lead to loss of combustion air and a stalling of the engine. Without crew onboard the ASCC would suffer this damage, stall, and be unable to continue its mission. This study took this particular event into account for analysis. A single mast submergence event on a 50 NM trip was considered unacceptable and can be seen in red in the following figures. Bow and fore quarter seas produce the worst results. Sea states above 3 in all configurations showed at least one event per trip at transit speeds over 15 knots.

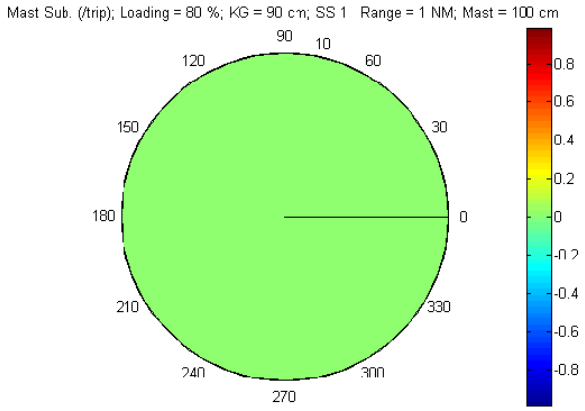


Figure 12. **Mast Submergence at 80% Load, Sea State 1, Midbox Load.**

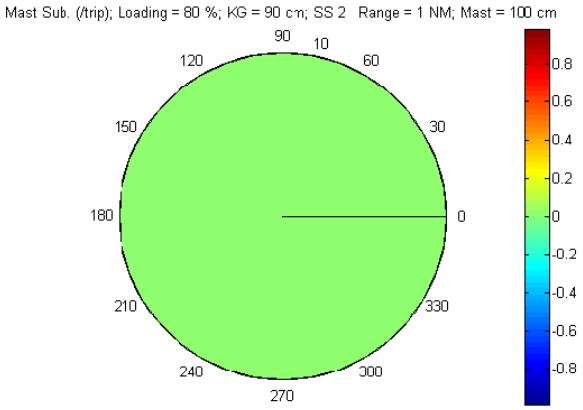


Figure 13. **Mast Submergence at 80% Load, Sea State 2, Midbox Load.**

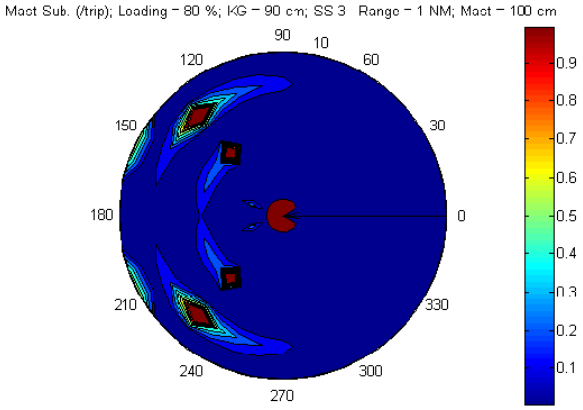


Figure 14. **Mast Submergence at 80% Load, Sea State 3, Midbox Load.**

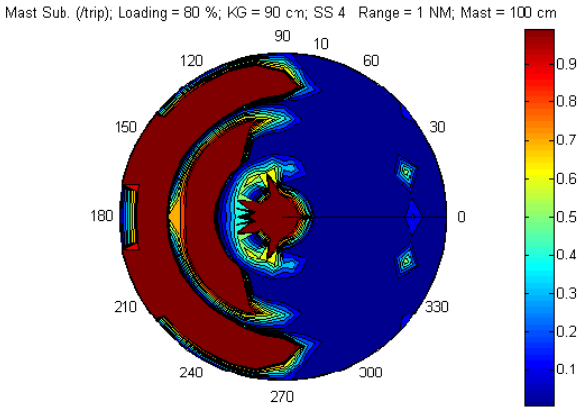


Figure 15. **Mast Submergence at 80% Load, Sea State 4, Midbox Load.**

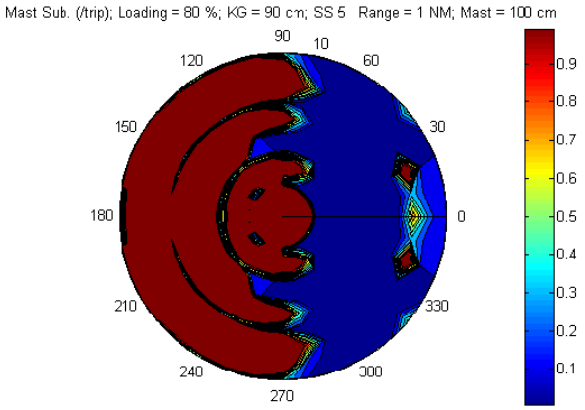


Figure 16. **Mast Submergence at 80% Load, Sea State 5, Midbox Load.**

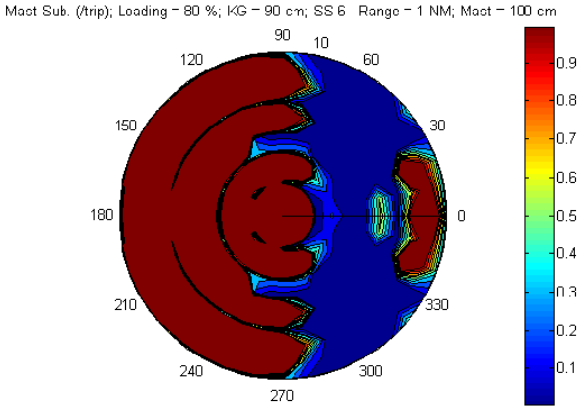


Figure 17. **Mast Submergence at 80% Load, Sea State 6, Midbox Load.**

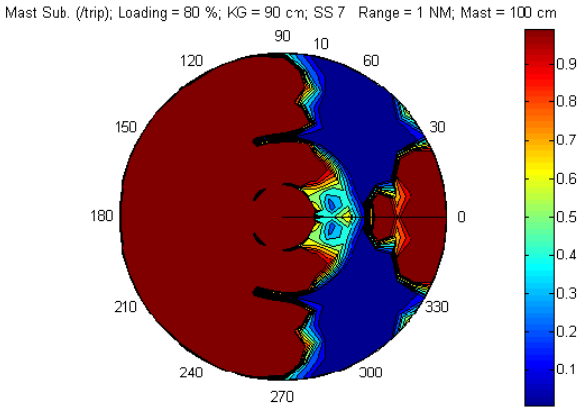


Figure 18. **Mast Submergence at 80% Load, Sea State 7, Midbox Load.**

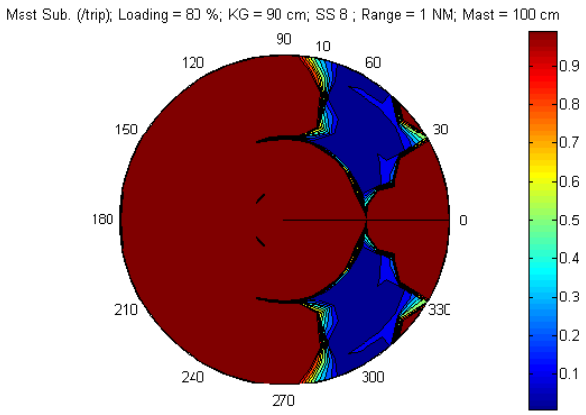


Figure 19. **Mast Submergence at 80% Load, Sea State 8, Midbox Load.**

b. Propeller Racing

As previously noted, the propeller racing events are noteworthy simply to show the coincidence of pitch and roll to overall ships operability. As with mast submergence, bow and fore quartering seas produce the most events and these start as low as sea state 3.

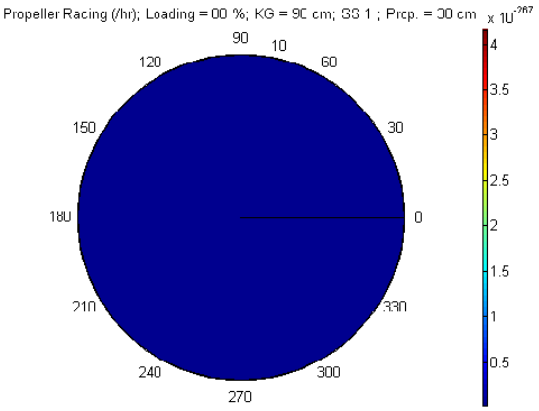


Figure 20. **Propeller racing, 30 cm above keel at 80% Load, Sea State 1, Midbox Load.**

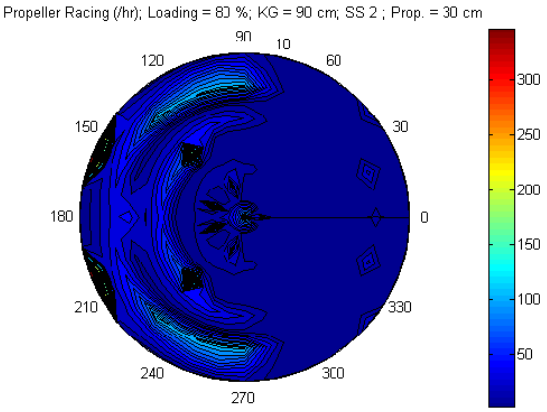


Figure 21. **Propeller racing, 30 cm above keel at 80% Load, Sea State 2, Midbox Load.**

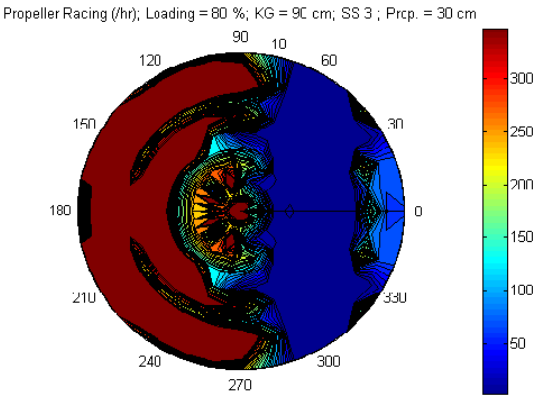


Figure 22. **Propeller racing, 30 cm above keel at 80% Load, Sea State 3, Midbox Load.**

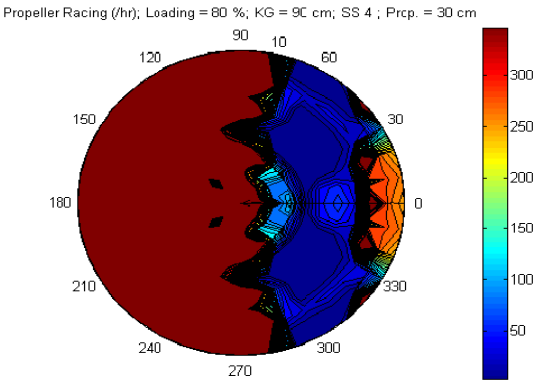


Figure 23. **Propeller racing, 30 cm above keel at 80% Load, Sea State 4, Midbox Load.**

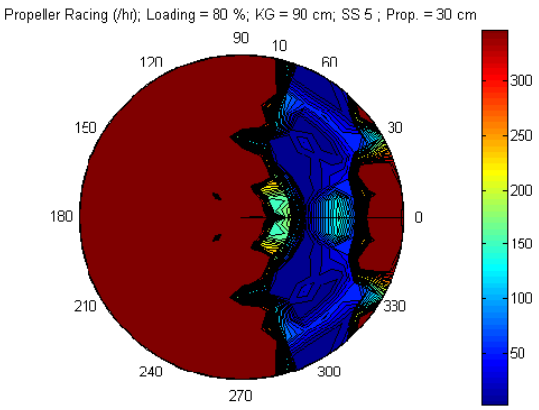


Figure 24. **Propeller racing, 30 cm above keel at 80% Load, Sea State 5, Midbox Load.**

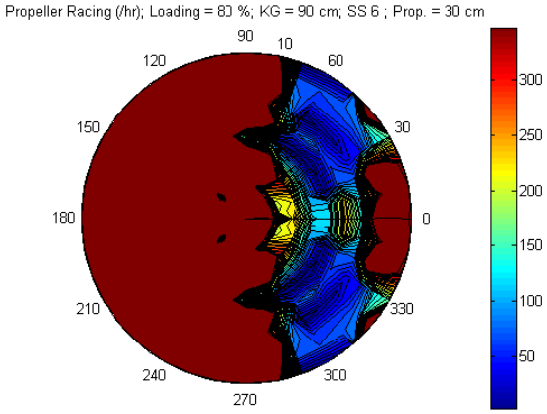


Figure 25. **Propeller racing, 30 cm above keel at 80% Load, Sea State 6, Midbox Load.**

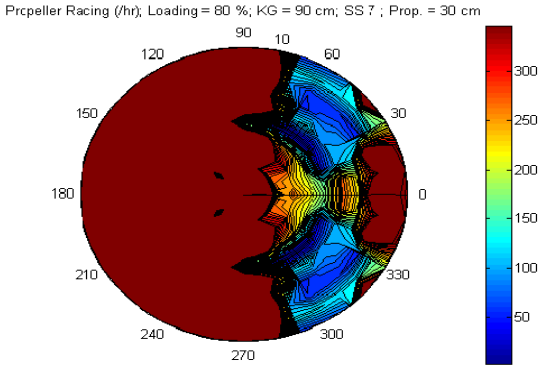


Figure 26. **Propeller racing, 30 cm above keel at 80% Load, Sea State 7, Midbox Load.**

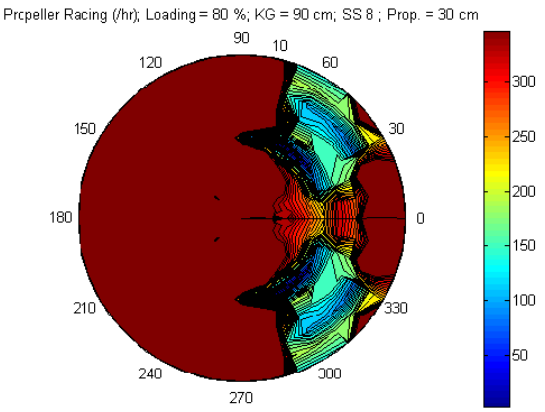


Figure 27. **Propeller racing, 30 cm above keel at 80% Load, Sea State 8, Midbox Load.**

By lowering the estimated propeller depth by 10 cm, the following figures are output. Although not readily obvious in this format, the change made had little effect on the likelihood of racing events. This will be addressed more thoroughly in a follow-on operability study.

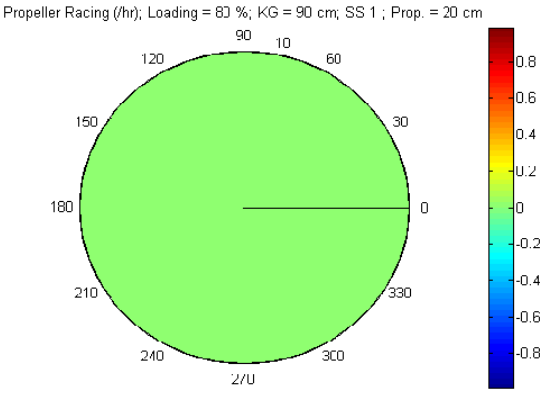


Figure 28. **Propeller racing, 20 cm above keel at 80% Load, Sea State 1, Midbox Load.**

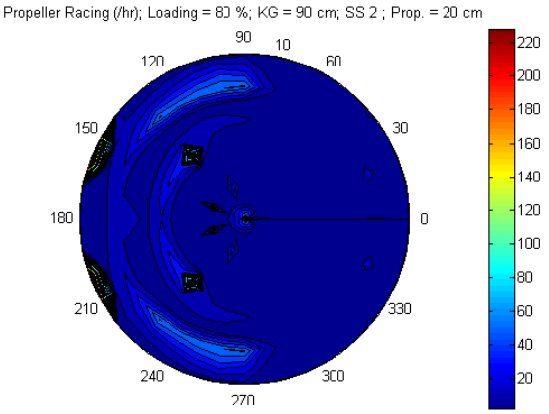


Figure 29. **Propeller racing, 20 cm above keel at 80% Load, Sea State 2, Midbox Load.**

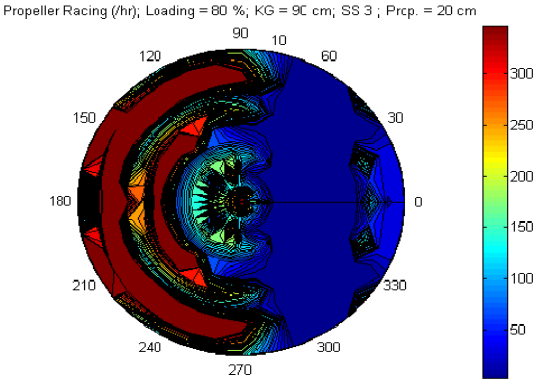


Figure 30. **Propeller racing, 20 cm above keel at 80% Load, Sea State 3, Midbox Load.**

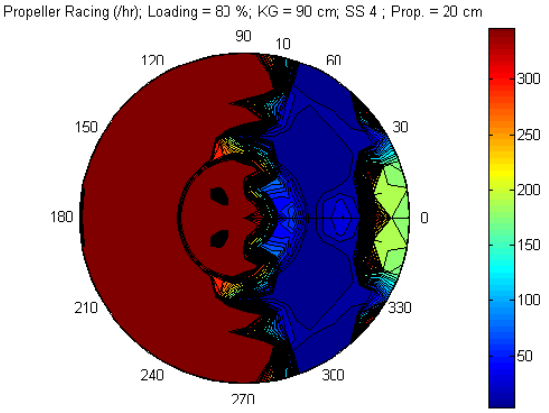


Figure 31. **Propeller racing, 20 cm above keel at 80% Load, Sea State 4, Midbox Load.**

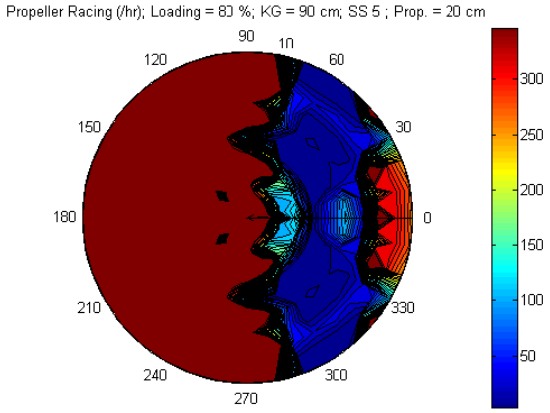


Figure 32. **Propeller racing, 20 cm above keel at 80% Load, Sea State 5, Midbox Load.**

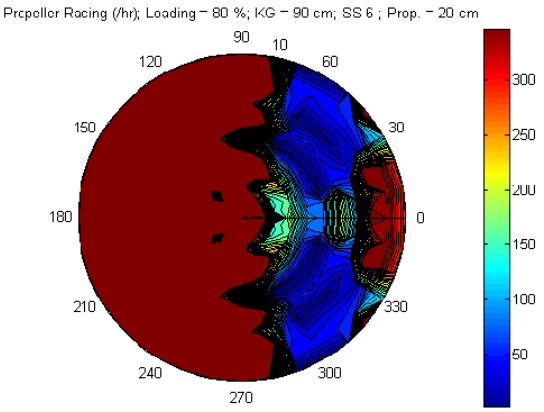


Figure 33. **Propeller racing, 20 cm above keel at 80% Load, Sea State 6, Midbox Load.**

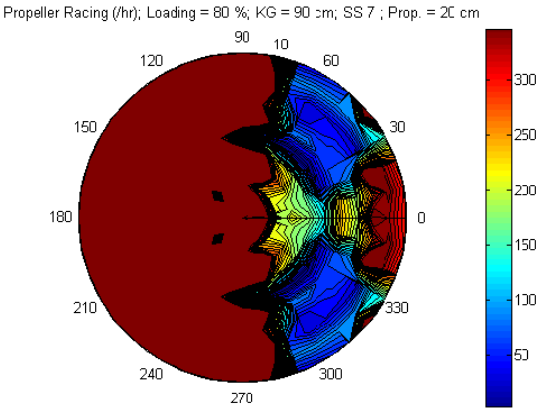


Figure 34. **Propeller racing, 20 cm above keel at 80% Load, Sea State 7, Midbox Load.**

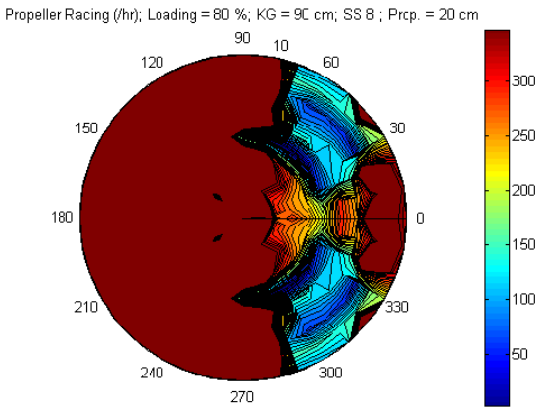


Figure 35. **Propeller racing, 20 cm above keel at 80% Load, Sea State 8, Midbox Load.**

c. Pitch and Roll

The results show a distinct interaction between pitch and roll and mast submergence events and propeller racing events. Note in the following figures that both pitch and roll degrees go in extremis with waves on the bow or the fore quarters. It is also noteworthy that where the red areas of the pitch and roll motions overlap, both mast submergence and propeller racing go red.

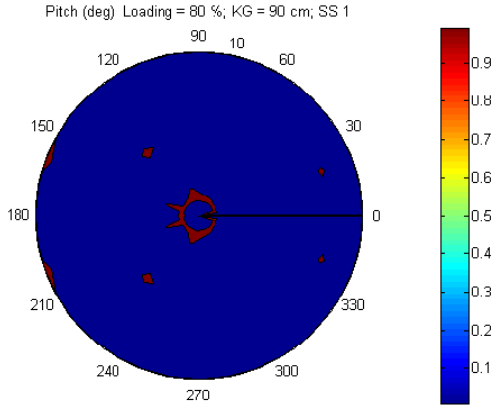


Figure 36. **Pitch at 80% Load, Sea State 1, Midbox Load.**

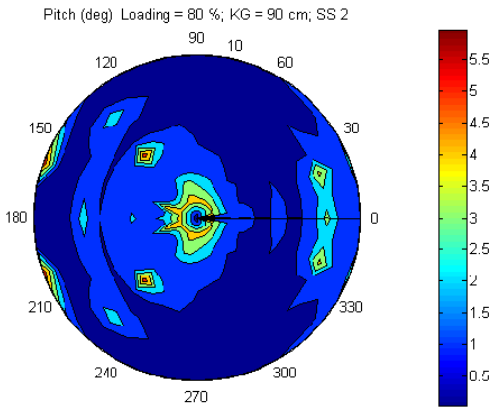


Figure 37. **Pitch at 80% Load, Sea State 2, Midbox Load.**

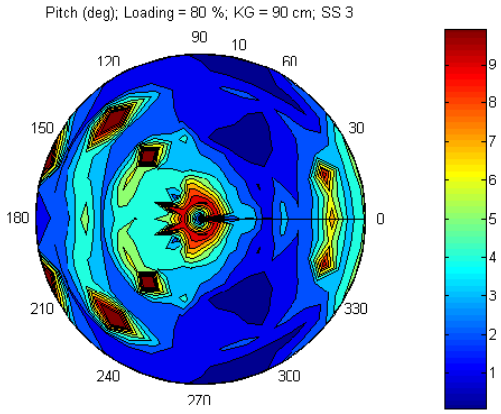


Figure 38. **Pitch at 80% Load, Sea State 3, Midbox Load.**

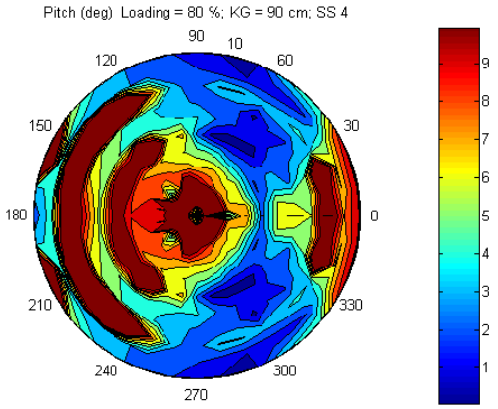


Figure 39. **Pitch at 80% Load, Sea State 4, Midbox Load.**

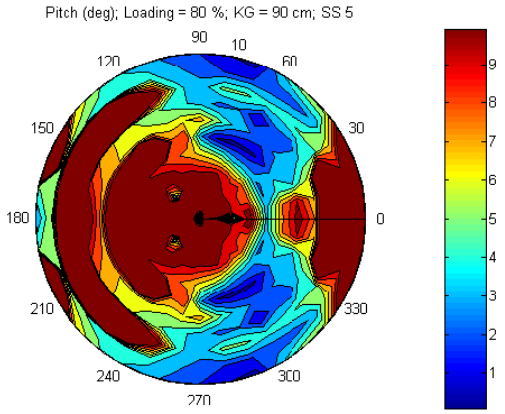


Figure 40. **Pitch at 80% Load, Sea State 5, Midbox Load.**

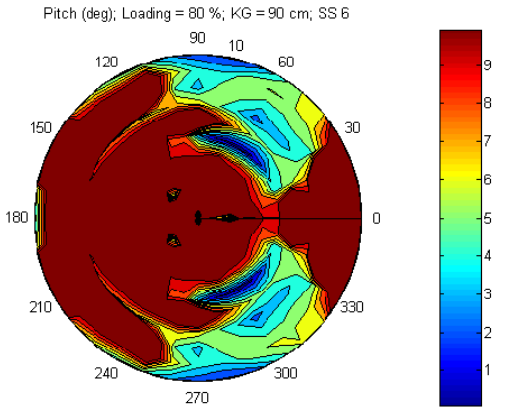


Figure 41. **Pitch at 80% Load, Sea State 6, Midbox Load.**

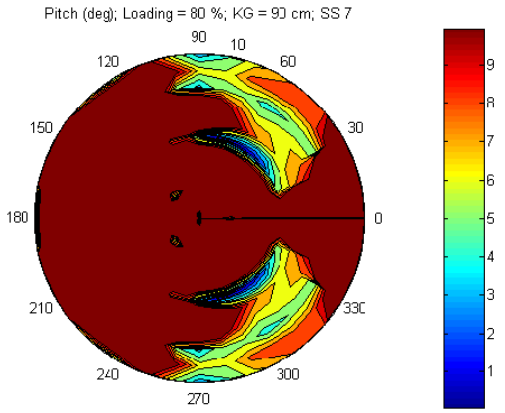


Figure 42. **Pitch at 80% Load, Sea State 7, Midbox Load.**

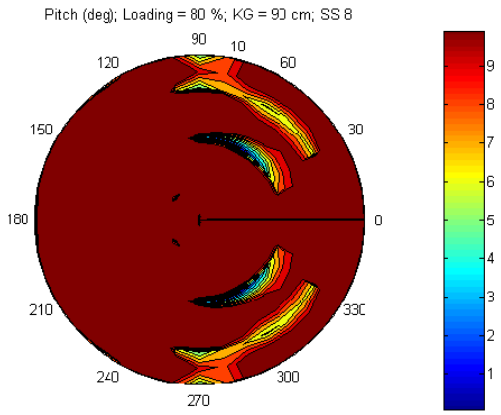


Figure 43. **Pitch at 80% Load, Sea State 8, Midbox Load.**

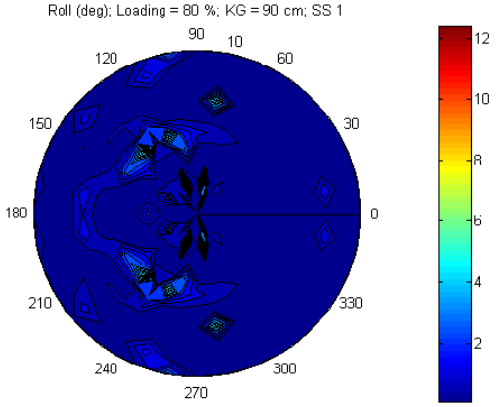


Figure 44. **Roll at 80% Load, Sea State 1, Midbox Load.**

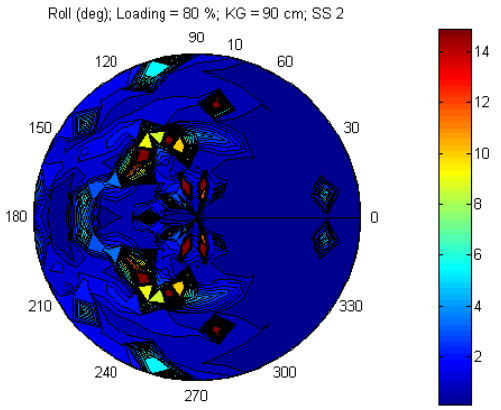


Figure 45. **Roll at 80% Load, Sea State 2, Midbox Load.**

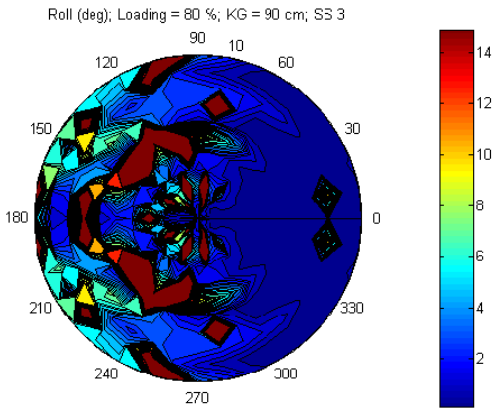


Figure 46. **Roll at 80% Load, Sea State 3, Midbox Load.**

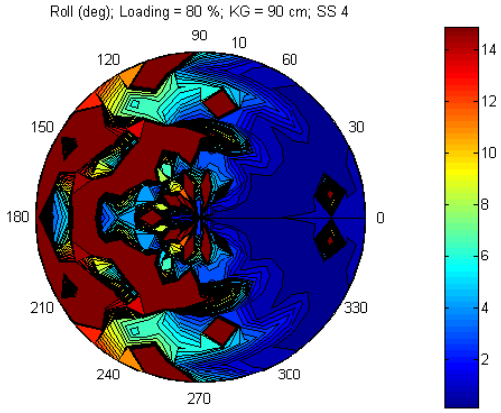


Figure 47. **Roll at 80% Load, Sea State 4, Midbox Load.**

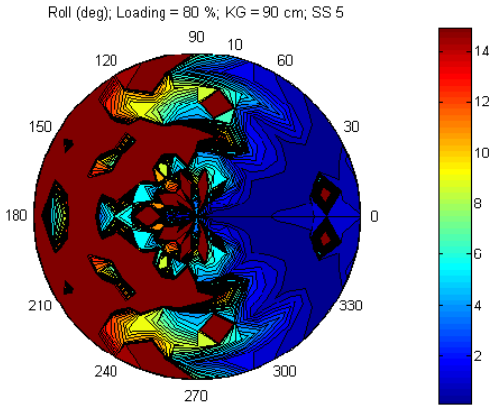


Figure 48. **Roll at 80% Load, Sea State 5, Midbox Load.**

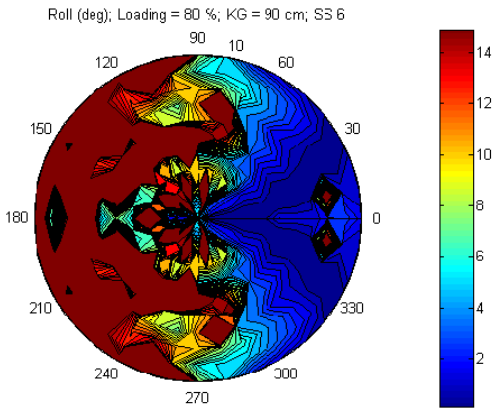


Figure 49. **Roll at 80% Load, Sea State 6, Midbox Load.**

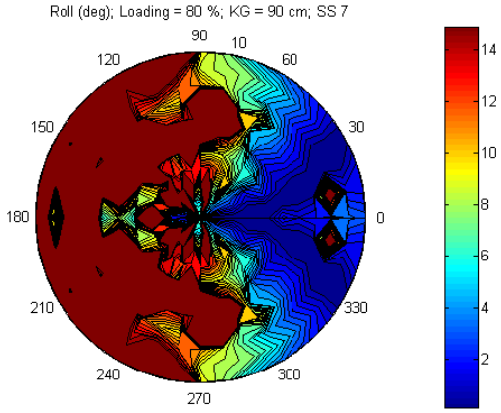


Figure 50. **Roll at 80% Load, Sea State 7, Midbox Load.**

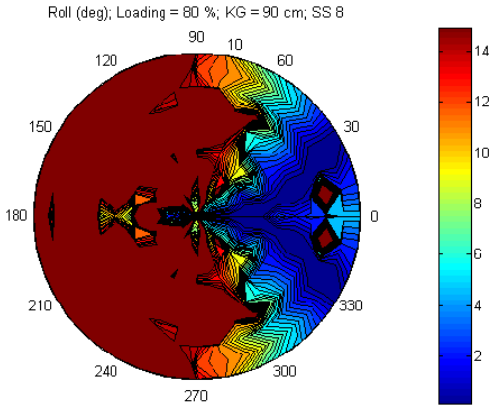


Figure 51. **Roll at 80% Load, Sea State 8, Midbox Load.**

D. CONCLUSSIONS

1. Design/Mission Shortcomings

While the goal of the ASCC is to operate in sea states above the current systems, Figures 52 and 53 indicate that this may not be entirely possible. By putting in a regional probability of occurrence for the different sea states and overlaying them with the operability of the vessel with all of the previously mentioned parametrics, it can be seen that only in sea states 1 and 2 is there a low occurrence of operational interruptions. At sea state 3 the ASCC would only be able to operate less than 65% of the time. At sea state 4, the set goal, operability drops below 45%. It must be noted that these figures are for all speeds and headings. It would be possible to increase the operability of the ASCC by mandating wave heading and slower speeds, but under the current design sea state 4 is simply too damaging to be able to successfully operate. Also note that very little difference is made between the results of the two assumed propeller height values.

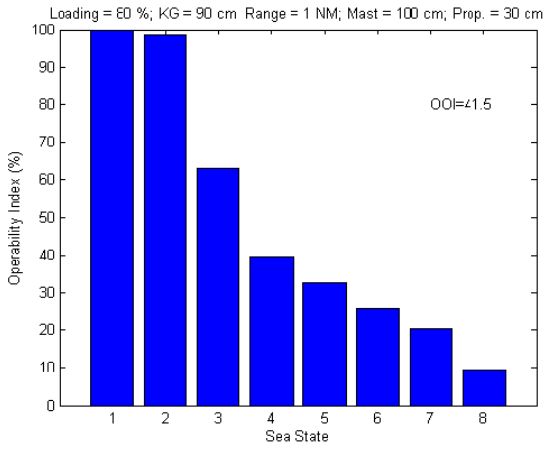


Figure 52. **Overall Operability Index for 80% Midbox Load and 30 cm propeller height.**

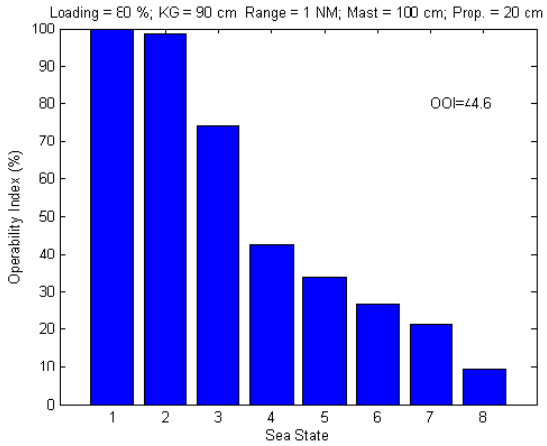


Figure 53. **Overall Operability Index for 80% Midbox Load and 20 cm propeller height.**

2. Improvement Possibilities

Although the predicted response is deemed satisfactory for the intended purpose of the ASCC, it seakeeping performance could be improved as follows: (1) Increased mast height in order to reduce mast submergence events. Shown below are mast submergence events for mast height configurations from 1 to 2 meters, 50% loaded (return load), a KG of 50 cm (upper midbox), and at the design goal of sea state 4. As can bee seen, raising the mast head height and engine intake significantly reduces the number of submergence events down to very rare for the 2 m case. This, however, might increase the center of gravity of the system and would need further testing. (2) Increased roll damping through rudder control. This, however, might increase the added resistance due to rudder action and, therefore, reduce the available payload. (3) Ballasted outriggers in order to decrease roll period. This, however, might increase resistance and complicate launch and recovery, and limit the number of containers that can be carried by the logistics ship.

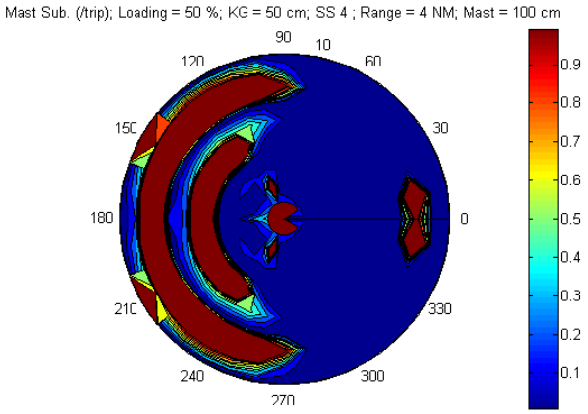


Figure 54. **Mast submergence at the initial mast height of 1 m.**

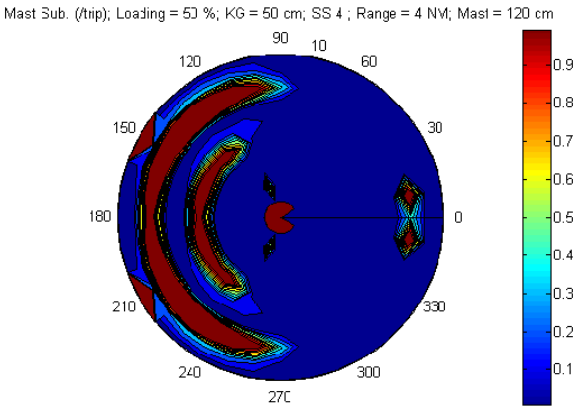


Figure 55. **Mast submergence with a mast height of 1.2 m.**

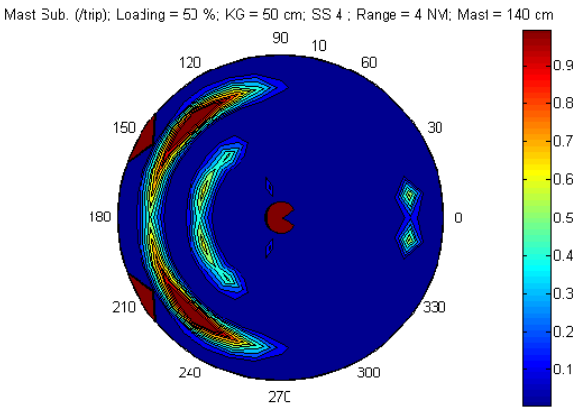


Figure 56. **Mast submergence with a mast height of 1.4 m.**

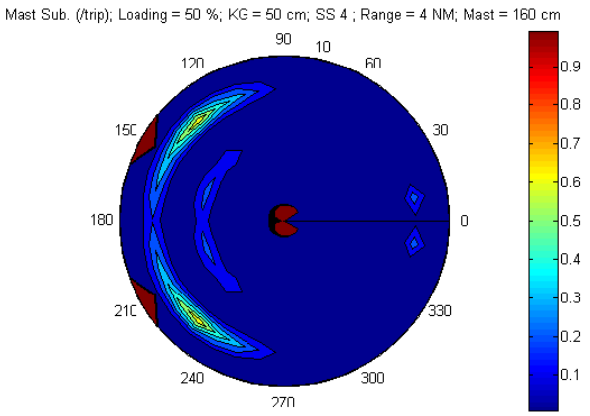


Figure 57. **Mast submergence with a mast height of 1.6 m.**

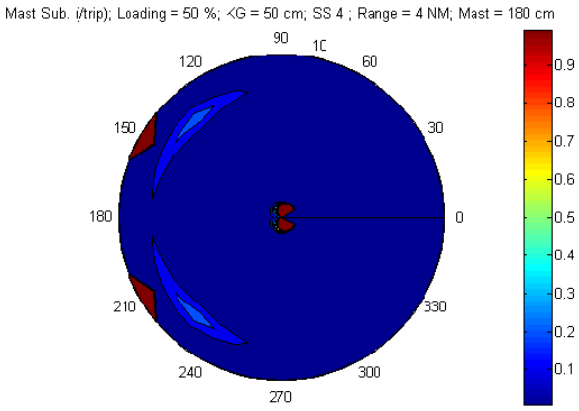


Figure 58. **Mast submergence with a mast height of 1.8 m.**

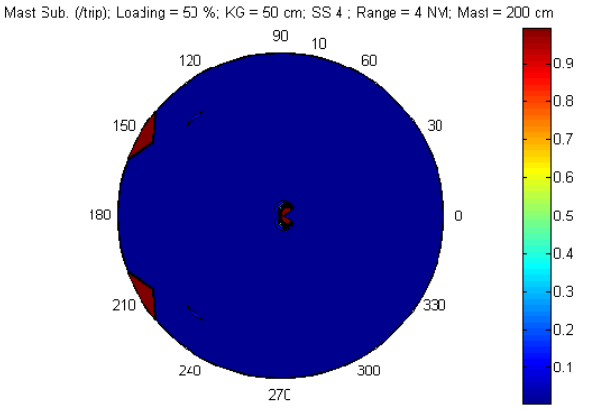


Figure 59. **Mast submergence with a mast height of 2 m.**

3. The Next Step

More detailed seakeeping calculations could be performed by taking into consideration the exact shape of the bow and stern units, along with experimental verification. Detailed graphical modeling would also be helpful in further determining root causes to the mast submergence and propeller racing events. Agencies such as the David Taylor Institute and Naval Surface Warfare Center Carderock could be engaged to conduct both digital FEM and CDF modeling and physical tank model experiments to further refine these results.

APPENDIX A. MATLAB COLLATION FUNCTION

```
function fcn_polar_speed_loop(Loading_Cond,KG,z_prop_keel,z_mast_top,range_NM)
% ASCC Container Data Set
% Horizontal/Vertical motions - Polar plots
% Speed in knots - Heading (180 head, 0 stern)
% Contour plots (heading/speed)
% Two parameter Bretchneider spectrum - Long crested seas
% Dimensional version (Metric units)
%
% General Loop - meant for overnight runs
%
warning off
%
% GENERAL DATA
%
lambda_min   =   5.000;       % Min wave length (m)
lambda_max   = 505.000;       % Max wave length (m)
delta_lambda =  10.060;       % Wave length increment (m)
rho          =1024.000;       % Water density
zeta         =   1.000;       % Regular wave height
L            =   6.058;       % Reference length for nondimensionalization
g            =   9.810;       % Gravitational constant
%
% Sea state information
%
sig_wave_height  = [0.1 0.4 1.1 2.4 3.5  5.5  8.0 12.0];
modal_period_sec = [6.3 6.5 7.5 8.8 9.7 12.4 15.0 16.5];
percent_probab   = [1 6 15 32 21 15 8 2];
%
% Set limits for operations (PNA Vol. III)
% All values in RMS single amplitudes
% at Point-of-Interest (POI) where applicable
%
c_p_roll_max    = 15;            % Roll (deg)
c_p_pitch_max   = 10;            % Pitch (deg)
c_p_ver_acc_max = 1.0;           % Abs. Vert. Accel. at amidships (g's)
c_p_lat_acc_max = 0.5;           % Abs. Lat. Accel. at amidships (g's)
c_p_bow_acc_max = 1.0;           % Slam Accel. at bow (g's)
c_p_bow_vel_max = 3.0;           % Relative vert. velocity at bow (m/s)
c_p_prop_max    = 350;           % Propeller racing events (per hour)
c_p_mast_trip_max    = 1;        % Mast submergence events (per trip)
%
c_p_roll    = [0:0.5:c_p_roll_max];
c_p_pitch   = [0:1:c_p_pitch_max];
c_p_ver_acc = [0:0.05:c_p_ver_acc_max];
c_p_lat_acc = [0:0.02:c_p_lat_acc_max];
c_p_bow_acc = [0:0.05:c_p_bow_acc_max];
c_p_bow_vel = [0:0.10:c_p_bow_vel_max];
c_p_prop    = [0:10:c_p_prop_max];
c_p_mast_trip  = [0:0.1:c_p_mast_trip_max];
%
c_p_roll_max_string    = num2str(c_p_roll_max);
c_p_pitch_max_string   = num2str(c_p_pitch_max);
c_p_ver_acc_max_string = num2str(c_p_ver_acc_max);
c_p_lat_acc_max_string = num2str(c_p_lat_acc_max);
c_p_bow_acc_max_string = num2str(c_p_bow_acc_max);
c_p_bow_vel_max_string = num2str(c_p_bow_vel_max);
c_p_prop_max_string    = num2str(c_p_prop_max);
c_p_mast_trip_max_string    = num2str(c_p_mast_trip_max);
%
% Speed range (m/sec)
%
V_min   = 0;
V_max   = 10;
delta_V = 1;
Speed_Range_string='All Speeds';
Loading_Cond_string = num2str(Loading_Cond);
```

```
if Loading_Cond == 80
    draft = 1.27;
end
if Loading_Cond == 70
    draft = 1.14;
end
if Loading_Cond == 60
    draft = 1.01;
end
if Loading_Cond == 50
    draft = 0.88;
end
%
KG_string = num2str(KG);
%
% Point for motions calculation
%
z_point = 0.00;
x_point = 0.00;
y_point = 0.00;
%
z_prop_keel_string = num2str(z_prop_keel);
z_prop = z_prop_keel/100 - draft;
z_prop_string = num2str(z_prop);
x_prop = -L/2;
y_prop = 0.00;
%
z_mast = z_mast_top/100 + 2.5 - draft;
z_mast_top_string = num2str(z_mast_top);
z_mast_string = num2str(z_mast);
x_mast = -L/2;
y_mast = 0.00;
%
range_string = num2str(range_NM);
%
%
% Loop over sea state
%
i_Sea_State=0;
for Sea_State=1:1:8,
i_Sea_State=i_Sea_State+1;
[Loading_Cond KG z_prop_keel z_mast_top range_NM Sea_State]
S_S_string = num2str(Sea_State);
HS=sig_wave_height(Sea_State);
HS_string=num2str(HS);
T_m=modal_period_sec(Sea_State);
T_m_string = num2str(T_m);
omega_m=2*pi/T_m;
%
% Set up file reading format.
%
trigg = 30;
f1loc = 25;
f2loc = 26;
f3loc = 27;
f4loc = 28;
f5loc = 29;
f6loc = 30;
%
lambda      = lambda_min:delta_lambda:lambda_max;          % Vector of wavelengths
lambdaoverL = lambda/L;
wavenumber  = 2.0*pi./lambda;                              % Wave number
omega       = sqrt(wavenumber*g);                          % Wave frequency
wavenumber  = wavenumber';
omega       = omega';
period      = 2.0*pi./omega;
filesize    = size(lambda);
lambda_size = trigg*filesize(2);
%
beta_incr  = 10;
%
```

```
iV=0;
for V        = V_min:delta_V:V_max,
    iV       = iV+1;
    V_string = num2str(V);
    V_calc   = V;    % m/sec
    V_knots  = V/0.5144;
    time_hrs = range_NM/(V_knots+0.01);
    ibeta    = 0;
    for beta = 0:beta_incr:360,       % Loop on sea direction
        if beta > 180
          beta=  360-beta;
        end
        beta_string = num2str(beta);
        beta_calc   = beta*pi/180;
        ibeta       = ibeta+1;
        %
        % Load data file V_beta.txt
        %
        load_filename =
strcat('results\',Loading_Cond_string,'_',KG_string,'_',V_string,'_',beta_string,'.txt');
        filename      = load(load_filename);
        %
        omegae     = omega-wavenumber*V_calc*cos(beta_calc);    % Frequency of encounter
        periode    = 2.0*pi./omegae;
        %
        % Set mass matrix elements
        %
        M11 = filename(1:trigg:lambda_size,1);
        M12 = filename(1:trigg:lambda_size,2);
        M13 = filename(1:trigg:lambda_size,3);
        M14 = filename(1:trigg:lambda_size,4);
        M15 = filename(1:trigg:lambda_size,5);
        M16 = filename(1:trigg:lambda_size,6);
        M21 = filename(2:trigg:lambda_size,1);
        M22 = filename(2:trigg:lambda_size,2);
        M23 = filename(2:trigg:lambda_size,3);
        M24 = filename(2:trigg:lambda_size,4);
        M25 = filename(2:trigg:lambda_size,5);
        M26 = filename(2:trigg:lambda_size,6);
        M31 = filename(3:trigg:lambda_size,1);
        M32 = filename(3:trigg:lambda_size,2);
        M33 = filename(3:trigg:lambda_size,3);
        M34 = filename(3:trigg:lambda_size,4);
        M35 = filename(3:trigg:lambda_size,5);
        M36 = filename(3:trigg:lambda_size,6);
        M41 = filename(4:trigg:lambda_size,1);
        M42 = filename(4:trigg:lambda_size,2);
        M43 = filename(4:trigg:lambda_size,3);
        M44 = filename(4:trigg:lambda_size,4);
        M45 = filename(4:trigg:lambda_size,5);
        M46 = filename(4:trigg:lambda_size,6);
        M51 = filename(5:trigg:lambda_size,1);
        M52 = filename(5:trigg:lambda_size,2);
        M53 = filename(5:trigg:lambda_size,3);
        M54 = filename(5:trigg:lambda_size,4);
        M55 = filename(5:trigg:lambda_size,5);
        M56 = filename(5:trigg:lambda_size,6);
        M61 = filename(6:trigg:lambda_size,1);
        M62 = filename(6:trigg:lambda_size,2);
        M63 = filename(6:trigg:lambda_size,3);
        M64 = filename(6:trigg:lambda_size,4);
        M65 = filename(6:trigg:lambda_size,5);
        M66 = filename(6:trigg:lambda_size,6);
        %
        M = [M11 M12 M13 M14 M15 M16;...
             M21 M22 M23 M24 M25 M26;...
             M31 M32 M33 M34 M35 M36;...
             M41 M42 M43 M44 M45 M46;...
             M51 M52 M53 M54 M55 M56;...
             M61 M62 M63 M64 M65 M66];
        %
```

```
% Added mass terms
%
A11 = filename(6+1:trigg:lambda_size,1);
A12 = filename(6+1:trigg:lambda_size,2);
A13 = filename(6+1:trigg:lambda_size,3);
A14 = filename(6+1:trigg:lambda_size,4);
A15 = filename(6+1:trigg:lambda_size,5);
A16 = filename(6+1:trigg:lambda_size,6);
A21 = filename(6+2:trigg:lambda_size,1);
A22 = filename(6+2:trigg:lambda_size,2);
A23 = filename(6+2:trigg:lambda_size,3);
A24 = filename(6+2:trigg:lambda_size,4);
A25 = filename(6+2:trigg:lambda_size,5);
A26 = filename(6+2:trigg:lambda_size,6);
A31 = filename(6+3:trigg:lambda_size,1);
A32 = filename(6+3:trigg:lambda_size,2);
A33 = filename(6+3:trigg:lambda_size,3);
A34 = filename(6+3:trigg:lambda_size,4);
A35 = filename(6+3:trigg:lambda_size,5);
A36 = filename(6+3:trigg:lambda_size,6);
A41 = filename(6+4:trigg:lambda_size,1);
A42 = filename(6+4:trigg:lambda_size,2);
A43 = filename(6+4:trigg:lambda_size,3);
A44 = filename(6+4:trigg:lambda_size,4);
A45 = filename(6+4:trigg:lambda_size,5);
A46 = filename(6+4:trigg:lambda_size,6);
A51 = filename(6+5:trigg:lambda_size,1);
A52 = filename(6+5:trigg:lambda_size,2);
A53 = filename(6+5:trigg:lambda_size,3);
A54 = filename(6+5:trigg:lambda_size,4);
A55 = filename(6+5:trigg:lambda_size,5);
A56 = filename(6+5:trigg:lambda_size,6);
A61 = filename(6+6:trigg:lambda_size,1);
A62 = filename(6+6:trigg:lambda_size,2);
A63 = filename(6+6:trigg:lambda_size,3);
A64 = filename(6+6:trigg:lambda_size,4);
A65 = filename(6+6:trigg:lambda_size,5);
A66 = filename(6+6:trigg:lambda_size,6);
%
A = [A11 A12 A13 A14 A15 A16;...
     A21 A22 A23 A24 A25 A26;...
     A31 A32 A33 A34 A35 A36;...
     A41 A42 A43 A44 A45 A46;...
     A51 A52 A53 A54 A55 A56;...
     A61 A62 A63 A64 A65 A66];
%
% Damping terms
%
B11 = filename(12+1:trigg:lambda_size,1);
B12 = filename(12+1:trigg:lambda_size,2);
B13 = filename(12+1:trigg:lambda_size,3);
B14 = filename(12+1:trigg:lambda_size,4);
B15 = filename(12+1:trigg:lambda_size,5);
B16 = filename(12+1:trigg:lambda_size,6);
B21 = filename(12+2:trigg:lambda_size,1);
B22 = filename(12+2:trigg:lambda_size,2);
B23 = filename(12+2:trigg:lambda_size,3);
B24 = filename(12+2:trigg:lambda_size,4);
B25 = filename(12+2:trigg:lambda_size,5);
B26 = filename(12+2:trigg:lambda_size,6);
B31 = filename(12+3:trigg:lambda_size,1);
B32 = filename(12+3:trigg:lambda_size,2);
B33 = filename(12+3:trigg:lambda_size,3);
B34 = filename(12+3:trigg:lambda_size,4);
B35 = filename(12+3:trigg:lambda_size,5);
B36 = filename(12+3:trigg:lambda_size,6);
B41 = filename(12+4:trigg:lambda_size,1);
B42 = filename(12+4:trigg:lambda_size,2);
B43 = filename(12+4:trigg:lambda_size,3);
B44 = filename(12+4:trigg:lambda_size,4);
B45 = filename(12+4:trigg:lambda_size,5);
```

```
B46 = filename(12+4:trigg:lambda_size,6);
B51 = filename(12+5:trigg:lambda_size,1);
B52 = filename(12+5:trigg:lambda_size,2);
B53 = filename(12+5:trigg:lambda_size,3);
B54 = filename(12+5:trigg:lambda_size,4);
B55 = filename(12+5:trigg:lambda_size,5);
B56 = filename(12+5:trigg:lambda_size,6);
B61 = filename(12+6:trigg:lambda_size,1);
B62 = filename(12+6:trigg:lambda_size,2);
B63 = filename(12+6:trigg:lambda_size,3);
B64 = filename(12+6:trigg:lambda_size,4);
B65 = filename(12+6:trigg:lambda_size,5);
B66 = filename(12+6:trigg:lambda_size,6);
%
B = [B11 B12 B13 B14 B15 B16;...
     B21 B22 B23 B24 B25 B26;...
     B31 B32 B33 B34 B35 B36;...
     B41 B42 B43 B44 B45 B46;...
     B51 B52 B53 B54 B55 B56;...
     B61 B62 B63 B64 B65 B66];
%
% Hydrostatic terms
%
C11 = filename(18+1:trigg:lambda_size,1);
C12 = filename(18+1:trigg:lambda_size,2);
C13 = filename(18+1:trigg:lambda_size,3);
C14 = filename(18+1:trigg:lambda_size,4);
C15 = filename(18+1:trigg:lambda_size,5);
C16 = filename(18+1:trigg:lambda_size,6);
C21 = filename(18+2:trigg:lambda_size,1);
C22 = filename(18+2:trigg:lambda_size,2);
C23 = filename(18+2:trigg:lambda_size,3);
C24 = filename(18+2:trigg:lambda_size,4);
C25 = filename(18+2:trigg:lambda_size,5);
C26 = filename(18+2:trigg:lambda_size,6);
C31 = filename(18+3:trigg:lambda_size,1);
C32 = filename(18+3:trigg:lambda_size,2);
C33 = filename(18+3:trigg:lambda_size,3);
C34 = filename(18+3:trigg:lambda_size,4);
C35 = filename(18+3:trigg:lambda_size,5);
C36 = filename(18+3:trigg:lambda_size,6);
C41 = filename(18+4:trigg:lambda_size,1);
C42 = filename(18+4:trigg:lambda_size,2);
C43 = filename(18+4:trigg:lambda_size,3);
C44 = filename(18+4:trigg:lambda_size,4);
C45 = filename(18+4:trigg:lambda_size,5);
C46 = filename(18+4:trigg:lambda_size,6);
C51 = filename(18+5:trigg:lambda_size,1);
C52 = filename(18+5:trigg:lambda_size,2);
C53 = filename(18+5:trigg:lambda_size,3);
C54 = filename(18+5:trigg:lambda_size,4);
C55 = filename(18+5:trigg:lambda_size,5);
C56 = filename(18+5:trigg:lambda_size,6);
C61 = filename(18+6:trigg:lambda_size,1);
C62 = filename(18+6:trigg:lambda_size,2);
C63 = filename(18+6:trigg:lambda_size,3);
C64 = filename(18+6:trigg:lambda_size,4);
C65 = filename(18+6:trigg:lambda_size,5);
C66 = filename(18+6:trigg:lambda_size,6);
%
C = [C11 C12 C13 C14 C15 C16;...
     C21 C22 C23 C24 C25 C26;...
     C31 C32 C33 C34 C35 C36;...
     C41 C42 C43 C44 C45 C46;...
     C51 C52 C53 C54 C55 C56;...
     C61 C62 C63 C64 C65 C66];
%
% Total exciting forces
%
F1_amp = filename(f1loc:trigg:lambda_size,5);
F2_amp = filename(f2loc:trigg:lambda_size,5);
```

```
        F3_amp = filename(f3loc:trigg:lambda_size,5);
        F4_amp = filename(f4loc:trigg:lambda_size,5);
        F5_amp = filename(f5loc:trigg:lambda_size,5);
        F6_amp = filename(f6loc:trigg:lambda_size,5);
        %
        F1_pha = filename(f1loc:trigg:lambda_size,6);
        F2_pha = filename(f2loc:trigg:lambda_size,6);
        F3_pha = filename(f3loc:trigg:lambda_size,6);
        F4_pha = filename(f4loc:trigg:lambda_size,6);
        F5_pha = filename(f5loc:trigg:lambda_size,6);
        F6_pha = filename(f6loc:trigg:lambda_size,6);
        %
        F1 = F1_amp.*exp(i*F1_pha.*pi/180.0);
        F2 = F2_amp.*exp(i*F2_pha.*pi/180.0);
        F3 = F3_amp.*exp(i*F3_pha.*pi/180.0);
        F4 = F4_amp.*exp(i*F4_pha.*pi/180.0);
        F5 = F5_amp.*exp(i*F5_pha.*pi/180.0);
        F6 = F6_amp.*exp(i*F6_pha.*pi/180.0);
        %
        F = [F1;F2;F3;F4;F5;F6];
        %
        % Calculate motions
        %
        A11bar = -(omegae.^2).*(M11 + A11) + i*omegae.*B11 + C11;
        A13bar = -(omegae.^2).*(M13 + A13) + i*omegae.*B13 + C13;
        A15bar = -(omegae.^2).*(M15 + A15) + i*omegae.*B15 + C15;
        A22bar = -(omegae.^2).*(M22 + A22) + i*omegae.*B22 + C22;
        A24bar = -(omegae.^2).*(M24 + A24) + i*omegae.*B24 + C24;
        A26bar = -(omegae.^2).*(M26 + A26) + i*omegae.*B26 + C26;
        A31bar = -(omegae.^2).*(M31 + A31) + i*omegae.*B31 + C31;
        A33bar = -(omegae.^2).*(M33 + A33) + i*omegae.*B33 + C33;
        A35bar = -(omegae.^2).*(M35 + A35) + i*omegae.*B35 + C35;
        A42bar = -(omegae.^2).*(M42 + A42) + i*omegae.*B42 + C42;
        A44bar = -(omegae.^2).*(M44 + A44) + i*omegae.*B44 + C44;
        A46bar = -(omegae.^2).*(M46 + A46) + i*omegae.*B46 + C46;
        A51bar = -(omegae.^2).*(M51 + A51) + i*omegae.*B51 + C51;
        A53bar = -(omegae.^2).*(M53 + A53) + i*omegae.*B53 + C53;
        A55bar = -(omegae.^2).*(M55 + A55) + i*omegae.*B55 + C55;
        A62bar = -(omegae.^2).*(M62 + A62) + i*omegae.*B62 + C62;
        A64bar = -(omegae.^2).*(M64 + A64) + i*omegae.*B64 + C64;
        A66bar = -(omegae.^2).*(M66 + A66) + i*omegae.*B66 + C66;
        %
        eta1_c = (A33bar.*A55bar - A35bar.*A53bar)./...
            (A11bar.*A33bar.*A55bar - A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar +
A31bar.*A15bar.*A53bar + A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F1 - ...
            (A13bar.*A55bar - A15bar.*A53bar)./(A11bar.*A33bar.*A55bar -
A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar + A31bar.*A15bar.*A53bar +
A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F3 + ...
            (A13bar.*A35bar - A15bar.*A33bar)./(A11bar.*A33bar.*A55bar -
A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar + A31bar.*A15bar.*A53bar +
A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F5;
        eta2_c = (A44bar.*A66bar - A46bar.*A64bar)./...
            (A22bar.*A44bar.*A66bar - A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar +
A42bar.*A26bar.*A64bar + A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F2 - ...
            (A24bar.*A66bar - A26bar.*A64bar)./(A22bar.*A44bar.*A66bar -
A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar + A42bar.*A26bar.*A64bar +
A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F4 + ...
            (A24bar.*A46bar - A26bar.*A44bar)./(A22bar.*A44bar.*A66bar -
A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar + A42bar.*A26bar.*A64bar +
A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F6;
        eta3_c = -(A31bar.*A55bar - A35bar.*A51bar)./...
            (A11bar.*A33bar.*A55bar - A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar +
A31bar.*A15bar.*A53bar + A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F1 + ...
            (A11bar.*A55bar - A15bar.*A51bar)./(A11bar.*A33bar.*A55bar -
A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar + A31bar.*A15bar.*A53bar +
A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F3 - ...
            (A11bar.*A35bar - A15bar.*A31bar)./(A11bar.*A33bar.*A55bar -
A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar + A31bar.*A15bar.*A53bar +
A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F5;
        eta4_c = -(A42bar.*A66bar - A46bar.*A62bar)./...
```

```
            (A22bar.*A44bar.*A66bar - A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar +
A42bar.*A26bar.*A64bar + A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F2 + ...
            (A22bar.*A66bar - A26bar.*A62bar)./(A22bar.*A44bar.*A66bar -
A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar + A42bar.*A26bar.*A64bar +
A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F4 - ...
            (A22bar.*A46bar - A26bar.*A42bar)./(A22bar.*A44bar.*A66bar -
A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar + A42bar.*A26bar.*A64bar +
A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F6;
        eta5_c = (A31bar.*A53bar - A33bar.*A51bar)./...
            (A11bar.*A33bar.*A55bar - A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar +
A31bar.*A15bar.*A53bar + A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F1 - ...
            (A11bar.*A53bar - A13bar.*A51bar)./(A11bar.*A33bar.*A55bar -
A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar + A31bar.*A15bar.*A53bar +
A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F3 + ...
            (A11bar.*A33bar - A13bar.*A31bar)./(A11bar.*A33bar.*A55bar -
A11bar.*A35bar.*A53bar - A31bar.*A13bar.*A55bar + A31bar.*A15bar.*A53bar +
A51bar.*A13bar.*A35bar - A51bar.*A15bar.*A33bar).*F5;
        eta6_c = (A42bar.*A64bar - A44bar.*A62bar)./...
            (A22bar.*A44bar.*A66bar - A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar +
A42bar.*A26bar.*A64bar + A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F2 - ...
            (A22bar.*A64bar - A24bar.*A62bar)./(A22bar.*A44bar.*A66bar -
A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar + A42bar.*A26bar.*A64bar +
A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F4 + ...
            (A22bar.*A44bar - A24bar.*A42bar)./(A22bar.*A44bar.*A66bar -
A22bar.*A46bar.*A64bar - A42bar.*A24bar.*A66bar + A42bar.*A26bar.*A64bar +
A62bar.*A24bar.*A46bar - A62bar.*A26bar.*A44bar).*F6;
        %
        % Absolute motions
        %
        xi1_point = eta1_c + eta5_c*z_point - eta6_c*y_point;          % forward at POI
        xi2_point = eta2_c + eta6_c*x_point - eta4_c*z_point;          % lateral at POI
        xi3_point = eta3_c - eta5_c*x_point + eta4_c*y_point;          % vertical at POI
        xi3_prop  = eta3_c - eta5_c*x_prop  + eta4_c*y_prop;           % vertical at prop
        xi3_mast  = eta3_c - eta5_c*x_mast  + eta4_c*y_mast;           % vertical at mast
        xi3_bow   = eta3_c - eta5_c*(L/2);                             % vertical at bow
        %
        % Relative motions/velocities/accelerations
        %
        xi3_bow_rel   = xi3_bow - zeta;               % relative vertical motion at bow
        vi3_bow_rel   = i*omegae.*xi3_bow_rel;        % relative vertical velocity at bow
        vi3_point     = i*omegae.*xi3_point;          % absolute vertical velocity at POI
        xi3_point_rel = xi3_point - zeta;             % relative vertical motion at POI
        vi3_point_rel = i*omegae.*xi3_point_rel;      % relative vertical velocity at POI
        ai3_point     = -i*(omegae.^2).*xi3_point;    % absolute vertical accel at POI
        ai2_point     = -i*(omegae.^2).*xi2_point;    % absolute lateral accel at POI
        vi3_bow       = i*omegae.*xi3_bow;            % absolute vertical velocity at bow
        ai3_bow       = -i*(omegae.^2).*xi3_bow;      % absolute vertical accel at bow
        xi3_prop_rel  = xi3_prop - zeta;              % relative vertical motion at prop
        xi3_mast_rel  = xi3_mast - zeta;              % relative vertical motion at mast
        %
        % Begin random wave calculations
        % Long crested seas; Bretschneider spectral formulation
        %
        % Sea spectrum
        %
        S  = (1.25/4)*((omega_m^4./omega.^5)*(HS^2)).*exp(-1.25*(omega_m^4./omega.^4));
        Se = S./abs((1-(2.0/g)*omega*V_calc*cos(beta_calc)));     % Convert S(w) to S(we)
        %
        % Define response spectra
        %
        Seta3_c          = ((abs(eta3_c)).^2).*Se;
        Seta4_c          = ((abs(eta4_c)).^2).*Se;
        Seta5_c          = ((abs(eta5_c)).^2).*Se;
        Sxi1_point       = ((abs(xi1_point)).^2).*Se;
        Sxi2_point       = ((abs(xi2_point)).^2).*Se;
        Sxi3_point       = ((abs(xi3_point)).^2).*Se;
        Sxi3_bow         = ((abs(xi3_bow)).^2).*Se;
        Sxi3_bow_rel     = ((abs(xi3_bow_rel)).^2).*Se;
        Svi3_point       = ((abs(vi3_point)).^2).*Se;
        Sai3_point       = ((abs(ai3_point)).^2).*Se;
        Sai2_point       = ((abs(ai2_point)).^2).*Se;
```

```
        Svi3_bow         = ((abs(vi3_bow)).^2).*Se;
        Svi3_bow_rel     = ((abs(vi3_bow_rel)).^2).*Se;
        Sai3_bow         = ((abs(ai3_bow)).^2).*Se;
        Svi3_point_rel   = ((abs(vi3_point_rel)).^2).*Se;
        Sxi3_point_rel   = ((abs(xi3_point_rel)).^2).*Se;
        Sxi3_point_rel_2 = (omegae.^2).*((abs(xi3_point_rel)).^2).*Se;
        Sxi3_prop_rel    = ((abs(xi3_prop_rel)).^2).*Se;
        Sxi3_prop_rel_2  = (omegae.^2).*((abs(xi3_prop_rel)).^2).*Se;
        Sxi3_mast_rel    = ((abs(xi3_mast_rel)).^2).*Se;
        Sxi3_mast_rel_2  = (omegae.^2).*((abs(xi3_mast_rel)).^2).*Se;
        %
        % Initializations
        %
        Seta3_c_i           = 0;
        Seta4_c_i           = 0;
        Seta5_c_i           = 0;
        Sxi1_point_i        = 0;
        Sxi2_point_i        = 0;
        Sxi3_point_i        = 0;
        Sxi3_bow_i          = 0;
        Sxi3_bow_rel_i      = 0;
        Svi3_point_i        = 0;
        Sai3_point_i        = 0;
        Sai2_point_i        = 0;
        Svi3_bow_i          = 0;
        Svi3_bow_rel_i      = 0;
        Sai3_bow_i          = 0;
        Svi3_point_rel_i    = 0;
        Sxi3_point_rel_i    = 0;
        Sxi3_point_rel_2_i  = 0;
        Sxi3_prop_rel_i     = 0;
        Sxi3_prop_rel_2_i   = 0;
        Sxi3_mast_rel_i     = 0;
        Sxi3_mast_rel_2_i   = 0;
        %
        % Integral S(w)*|RAO|^2
        %
        for I=2:1:filesize(2),
            Seta3_c_i        = Seta3_c_i        + 0.5*(Seta3_c(I)        + Seta3_c(I-1))
* abs((omegae(I-1)-omegae(I)));
            Seta4_c_i        = Seta4_c_i        + 0.5*(Seta4_c(I)        + Seta4_c(I-1))
* abs((omegae(I-1)-omegae(I)));
            Seta5_c_i        = Seta5_c_i        + 0.5*(Seta5_c(I)        + Seta5_c(I-1))
* abs((omegae(I-1)-omegae(I)));
            Sxi1_point_i     = Sxi1_point_i     + 0.5*(Sxi1_point(I)     + Sxi1_point(I-
1))     * abs((omegae(I-1)-omegae(I)));
            Sxi2_point_i     = Sxi2_point_i     + 0.5*(Sxi2_point(I)     + Sxi2_point(I-
1))     * abs((omegae(I-1)-omegae(I)));
            Sxi3_point_i     = Sxi3_point_i     + 0.5*(Sxi3_point(I)     + Sxi3_point(I-
1))     * abs((omegae(I-1)-omegae(I)));
            Sxi3_bow_i       = Sxi3_bow_i       + 0.5*(Sxi3_bow(I)       + Sxi3_bow(I-1))
* abs((omegae(I-1)-omegae(I)));
            Sxi3_bow_rel_i   = Sxi3_bow_rel_i   + 0.5*(Sxi3_bow_rel(I)   +
Sxi3_bow_rel(I-1))   * abs((omegae(I-1)-omegae(I)));
            Svi3_point_i     = Svi3_point_i     + 0.5*(Svi3_point(I)     + Svi3_point(I-
1))     * abs((omegae(I-1)-omegae(I)));
            Sai3_point_i     = Sai3_point_i     + 0.5*(Sai3_point(I)     + Sai3_point(I-
1))     * abs((omegae(I-1)-omegae(I)));
            Sai2_point_i     = Sai2_point_i     + 0.5*(Sai2_point(I)     + Sai2_point(I-
1))     * abs((omegae(I-1)-omegae(I)));
            Sai3_bow_i       = Sai3_bow_i       + 0.5*(Sai3_bow(I)       + Sai3_bow(I-1))
* abs((omegae(I-1)-omegae(I)));
            Svi3_bow_i       = Svi3_bow_i       + 0.5*(Svi3_bow(I)       + Svi3_bow(I-1))
* abs((omegae(I-1)-omegae(I)));
            Svi3_bow_rel_i   = Svi3_bow_rel_i   + 0.5*(Svi3_bow_rel(I)   +
Svi3_bow_rel(I-1))   * abs((omegae(I-1)-omegae(I)));
            Svi3_point_rel_i = Svi3_point_rel_i + 0.5*(Svi3_point_rel(I) +
Svi3_point_rel(I-1)) * abs((omegae(I-1)-omegae(I)));
            Sxi3_point_rel_i = Sxi3_point_rel_i + 0.5*(Sxi3_point_rel(I) +
Sxi3_point_rel(I-1)) * abs((omegae(I-1)-omegae(I)));
```

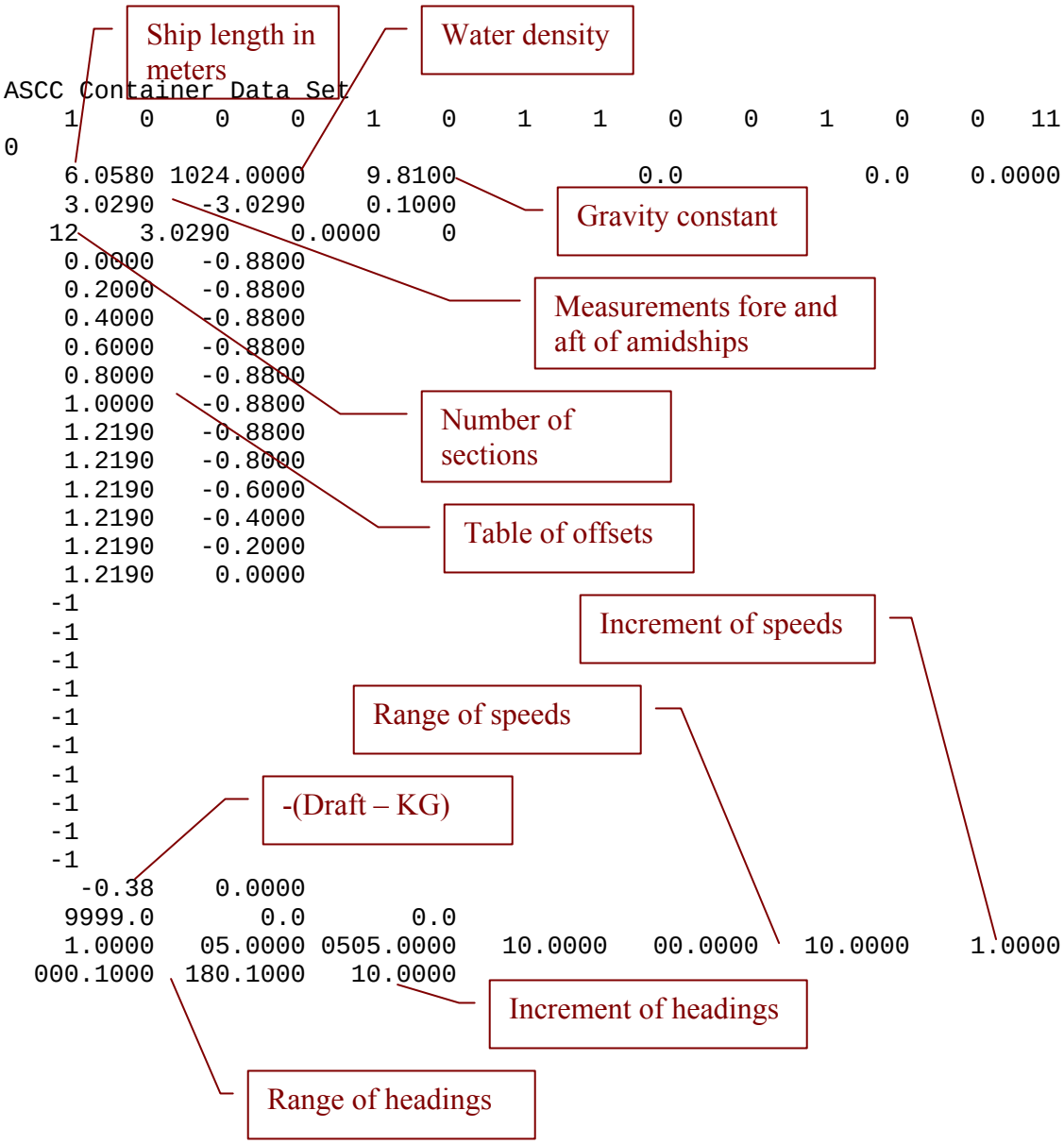
```
            Sxi3_prop_rel_i  = Sxi3_prop_rel_i  + 0.5*(Sxi3_prop_rel(I)  +
Sxi3_prop_rel(I-1))  * abs((omegae(I-1)-omegae(I)));
            Sxi3_mast_rel_i  = Sxi3_mast_rel_i  + 0.5*(Sxi3_mast_rel(I)  +
Sxi3_mast_rel(I-1))  * abs((omegae(I-1)-omegae(I)));
        end
        %
        % Second moment m_2
        %
        for I=2:1:filesize(2),
            Sxi3_point_rel_2_i = Sxi3_point_rel_2_i + 0.5*(Sxi3_point_rel_2(I) +
Sxi3_point_rel_2(I-1)) * abs((omegae(I-1)-omegae(I)));
            Sxi3_prop_rel_2_i  = Sxi3_prop_rel_2_i  + 0.5*(Sxi3_prop_rel_2(I)  +
Sxi3_prop_rel_2(I-1))  * abs((omegae(I-1)-omegae(I)));
            Sxi3_mast_rel_2_i  = Sxi3_mast_rel_2_i  + 0.5*(Sxi3_mast_rel_2(I)  +
Sxi3_mast_rel_2(I-1))  * abs((omegae(I-1)-omegae(I)));
        end
        %
        % RMS values
        %
        RMS_eta3_c          = sqrt(abs(Seta3_c_i));
        RMS_eta4_c          = 57.3*sqrt(abs(Seta4_c_i));
        RMS_eta5_c          = 57.3*sqrt(abs(Seta5_c_i));
        RMS_xi1_point       = sqrt(abs(Sxi1_point_i));
        RMS_xi2_point       = sqrt(abs(Sxi2_point_i));
        RMS_xi3_point       = sqrt(abs(Sxi3_point_i));
        RMS_xi3_bow         = sqrt(abs(Sxi3_bow_i));
        RMS_vi3_point       = sqrt(abs(Svi3_point_i));
        RMS_ai3_point       = sqrt(abs(Sai3_point_i));
        RMS_ai2_point       = sqrt(abs(Sai2_point_i));
        RMS_vi3_bow         = sqrt(abs(Svi3_bow_i));
        RMS_ai3_bow         = sqrt(abs(Sai3_bow_i));
        RMS_vi3_bow_rel     = sqrt(abs(Svi3_bow_rel_i));
        RMS_xi3_bow_rel     = sqrt(abs(Sxi3_bow_rel_i));
        RMS_vi3_point_rel   = sqrt(abs(Svi3_point_rel_i));
        RMS_xi3_point_rel   = sqrt(abs(Sxi3_point_rel_i));
        RMS_xi3_point_rel_2 = sqrt(abs(Sxi3_point_rel_2_i));
        RMS_xi3_prop_rel    = sqrt(abs(Sxi3_prop_rel_i));
        RMS_xi3_prop_rel_2  = sqrt(abs(Sxi3_prop_rel_2_i));
        RMS_xi3_mast_rel    = sqrt(abs(Sxi3_mast_rel_i));
        RMS_xi3_mast_rel_2  = sqrt(abs(Sxi3_mast_rel_2_i));
        %
        RMS_eta3_c_vector(ibeta,iV)          = RMS_eta3_c;
        RMS_eta4_c_vector(ibeta,iV)          = RMS_eta4_c;
        RMS_eta5_c_vector(ibeta,iV)          = RMS_eta5_c;
        RMS_xi1_point_vector(ibeta,iV)       = RMS_xi1_point;
        RMS_xi2_point_vector(ibeta,iV)       = RMS_xi2_point;
        RMS_xi3_point_vector(ibeta,iV)       = RMS_xi3_point;
        RMS_xi3_bow_vector(ibeta,iV)         = RMS_xi3_bow;
        RMS_vi3_point_vector(ibeta,iV)       = RMS_vi3_point;
        RMS_ai3_point_vector(ibeta,iV)       = RMS_ai3_point;
        RMS_ai2_point_vector(ibeta,iV)       = RMS_ai2_point;
        RMS_vi3_bow_vector(ibeta,iV)         = RMS_vi3_bow;
        RMS_vi3_bow_rel_vector(ibeta,iV)     = RMS_vi3_bow_rel;
        RMS_xi3_bow_rel_vector(ibeta,iV)     = RMS_xi3_bow_rel;
        RMS_ai3_bow_vector(ibeta,iV)         = RMS_ai3_bow;
        RMS_vi3_point_rel_vector(ibeta,iV)   = RMS_vi3_point_rel;
        RMS_xi3_point_rel_vector(ibeta,iV)   = RMS_xi3_point_rel;
        RMS_xi3_point_rel_2_vector(ibeta,iV) = RMS_xi3_point_rel_2;
        RMS_xi3_prop_rel_vector(ibeta,iV)    = RMS_xi3_prop_rel;
        RMS_xi3_prop_rel_2_vector(ibeta,iV)  = RMS_xi3_prop_rel_2;
        RMS_xi3_mast_rel_vector(ibeta,iV)    = RMS_xi3_mast_rel;
        RMS_xi3_mast_rel_2_vector(ibeta,iV)  = RMS_xi3_mast_rel_2;
        %

N_P_prop(ibeta,iV)=3600*(1/sqrt(2*pi))*(RMS_xi3_prop_rel_2/RMS_xi3_prop_rel)*exp(-
2*z_prop^2/(2*RMS_xi3_prop_rel));

N_P_mast(ibeta,iV)=time_hrs*3600*(1/sqrt(2*pi))*(RMS_xi3_mast_rel_2/RMS_xi3_mast_rel)*exp
(-2*z_mast^2/(2*RMS_xi3_mast_rel));
    end
end
```

```
lL=lambdaoverL;
%
temp_str = num2str(abs(str2num(beta_string)-180));
%
% Roll
%
figure(1)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_eta4_c_vector,c_p_roll),colorbar
title(['Roll (deg); Loading = ' Loading_Cond_string ' %; KG = ' KG_string ' cm; SS '
S_S_string])
figure_string=['graphs\roll\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'.emf'];
saveas(gcf,figure_string,'emf');
close
%
% Pitch
%
figure(2)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_eta5_c_vector,c_p_pitch),colorbar
title(['Pitch (deg); Loading = ' Loading_Cond_string ' %; KG = ' KG_string ' cm; SS '
S_S_string])
figure_string=['graphs\pitch\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'.emf'];
saveas(gcf,figure_string,'emf');
close
%
% Absolute Vertical Acceleration at POI
%
figure(3)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_ai3_point_vector/g,c_p_ver_acc),colorbar
title(['Absolute Vertical Acceleration (g); Loading = ' Loading_Cond_string ' %; KG = '
KG_string ' cm; SS ' S_S_string])
figure_string=['graphs\ver_acc\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'.emf']
;
saveas(gcf,figure_string,'emf');
close
%
% Absolute Lateral Acceleration at POI
%
figure(4)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_ai2_point_vector/g,c_p_lat_acc),colorbar
title(['Absolute Lateral Acceleration (g); Loading = ' Loading_Cond_string ' %; KG = '
KG_string ' cm; SS ' S_S_string])
figure_string=['graphs\lat_acc\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'.emf']
;
saveas(gcf,figure_string,'emf');
close
%
% Slam Velocity at bow
%
figure(5)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_vi3_bow_rel_vector,c_p_bow_vel),colorbar
```

```
title(['Slam Velocity (m/sec); Loading = ' Loading_Cond_string ' %; KG = ' KG_string '
cm; SS ' S_S_string])
figure_string=['graphs\slam_vel\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'.emf'
];
saveas(gcf,figure_string,'emf');
close
%
% Slam Acceleration at bow
%
figure(6)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_ai3_bow_vector/g,c_p_bow_acc),colorbar
title(['Slam Acceleration (g); Loading = ' Loading_Cond_string ' %; KG = ' KG_string '
cm; SS ' S_S_string])
figure_string=['graphs\slam_acc\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'.emf'
];
saveas(gcf,figure_string,'emf');
close
%
% Propeller Racing Events
%
figure(7)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',N_P_prop,c_p_prop),colorbar
title(['Propeller Racing (/hr); Loading = ' Loading_Cond_string ' %; KG = ' KG_string '
cm; SS ' S_S_string ' ; Prop. = ' z_prop_keel_string ' cm'])
figure_string=['graphs\race\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'_',z_prop
_keel_string,'.emf'];
saveas(gcf,figure_string,'emf');
close
%
% Mast Submergence Events
%
figure(8)
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',N_P_mast,c_p_mast_trip),colorbar
title(['Mast Sub. (/trip); Loading = ' Loading_Cond_string ' %; KG = ' KG_string ' cm; SS
' S_S_string ' ; Range = ' range_string ' NM; Mast = ' z_mast_top_string ' cm' ])
figure_string=['graphs\mast\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'_',range_
string,'_',z_mast_top_string,'.emf'];
saveas(gcf,figure_string,'emf');
close
%
% Operating Envelopes (for area calcs and OI processing)
%
figure(9)
colormap([0 0 0])
[th,r]=meshgrid((0:beta_incr:360)*pi/180,V_min:delta_V:V_max);
[X,Y]=pol2cart(th,r);
h=polar(th,r);delete(h);
hold on
contourf(X',Y',RMS_eta4_c_vector,[c_p_roll_max c_p_roll_max])
contourf(X',Y',RMS_eta5_c_vector,[c_p_pitch_max c_p_pitch_max])
contourf(X',Y',RMS_ai3_point_vector/g,[c_p_ver_acc_max c_p_ver_acc_max])
contourf(X',Y',RMS_ai2_point_vector/g,[c_p_lat_acc_max c_p_lat_acc_max])
contourf(X',Y',RMS_vi3_bow_rel_vector,[c_p_bow_vel_max c_p_bow_vel_max])
contourf(X',Y',RMS_ai3_bow_vector/g,[c_p_bow_acc_max c_p_bow_acc_max])
contourf(X',Y',N_P_prop,[c_p_prop_max c_p_prop_max])
contourf(X',Y',N_P_mast,[c_p_mast_trip_max c_p_mast_trip_max])
title(['OI; Loading = ' Loading_Cond_string ' %; KG = ' KG_string ' cm; SS ' S_S_string
'; Range = ' range_string ' NM; Mast = ' z_mast_top_string ' cm; Prop. = '
z_prop_keel_string ' cm'])
```

```
figure_string=['graphs\OI\',Loading_Cond_string,'_',KG_string,'_',S_S_string,'_',range_st
ring,'_',z_mast_top_string,'_',z_prop_keel_string,'.bmp'];
saveas(gcf,figure_string,'bmp');
fig1_data=imread(figure_string);
sum1_data=sum(sum(fig1_data));
hold off
y2_data=0;
y3_data=1;
x2_data=41904922;
x3_data=59138077;
x_data=sum1_data;
a_data=(y3_data-y2_data)/(x3_data-x2_data);
b_data=(y2_data*x3_data-y3_data*x2_data)/(x3_data-x2_data);
y_data=a_data*x_data+b_data;
y_data=y_data+0.01;
if y_data > 1
    y_data=1;
end
if y_data < 0
    y_data=0;
end
OI=100*y_data;
OI_string = strcat('OI=',num2str(OI,3),'%');
%text(-17,10,OI_string)
OI_vector(i_Sea_State)=100*y_data;
clear fig1_data
close
%
% Sea state loop ends
%
end
%
% Compute overall operability index
%
OOI = 100*sum((OI_vector/100).*(percent_probab/100));
OOI_string = strcat('OOI=',num2str(OOI,3));
figure(10)
colormap([0 0 1]),bar(OI_vector),xlabel('Sea State'),ylabel('Operability Index (%)')
axis([0 9 0 100])
title(['Loading = ' Loading_Cond_string ' %; KG = ' KG_string ' cm; Range = '
range_string ' NM; Mast = ' z_mast_top_string ' cm; Prop. = ' z_prop_keel_string ' cm'])
text(7,80,OOI_string)
figure_string=['graphs\OOI\',Loading_Cond_string,'_',KG_string,'_',range_string,'_',z_mas
t_top_string,'_',z_prop_keel_string,'.bmp'];
saveas(gcf,figure_string,'bmp');
close
clear all
```

APPENDIX B. SHIPMO SAMPLE INPUT FILE




```
ASCC Container Data Set
    1     0     0     0     1     0     1     1     0     0     1     0     0   11
0
    6.0580 1024.0000    9.8100          0.0          0.0    0.0000
    3.0290   -3.0290    0.1000
   12    3.0290    0.0000    0
    0.0000   -0.8800
    0.2000   -0.8800
    0.4000   -0.8800
    0.6000   -0.8800
    0.8000   -0.8800
    1.0000   -0.8800
    1.2190   -0.8800
    1.2190   -0.8000
    1.2190   -0.6000
    1.2190   -0.4000
    1.2190   -0.2000
    1.2190    0.0000
   -1
   -1
   -1
   -1
   -1
   -1
   -1
   -1
   -1
   -1
    -0.38    0.0000
    9999.0       0.0       0.0
    1.0000   05.0000 0505.0000   10.0000   00.0000   10.0000    1.0000
  000.1000  180.1000   10.0000
```

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. NONCRITICAL OUTPUT RESULTS

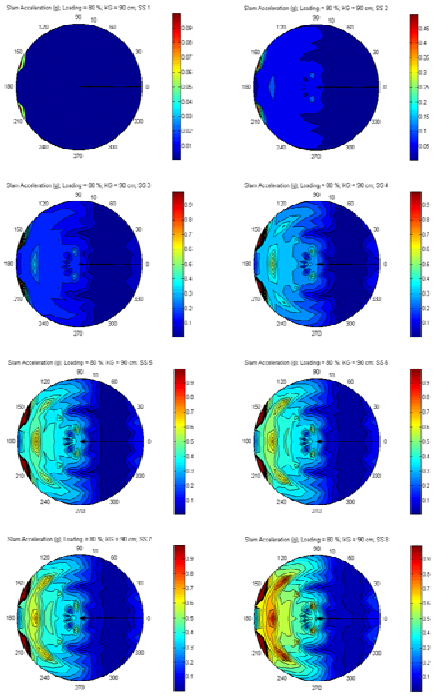


Figure 60. **Slam acceleration at 80% midbox load.**

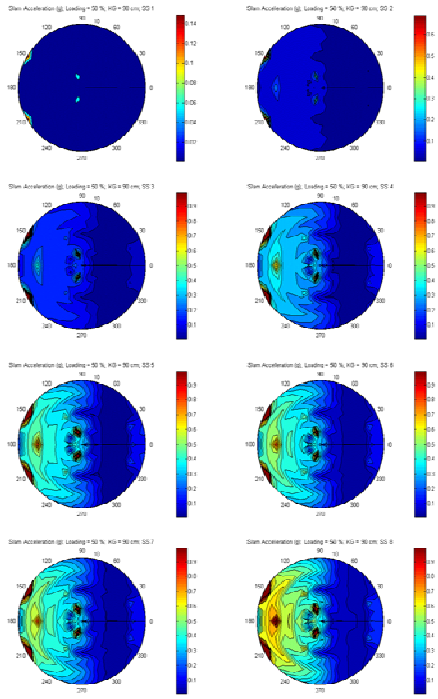


Figure 61. **Slam acceleration at 50% midbox load.**

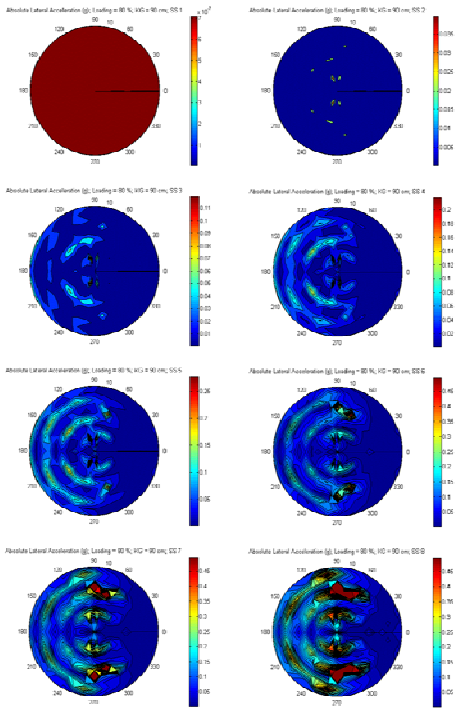


Figure 62. **Lateral acceleration at 80% midbox load.**

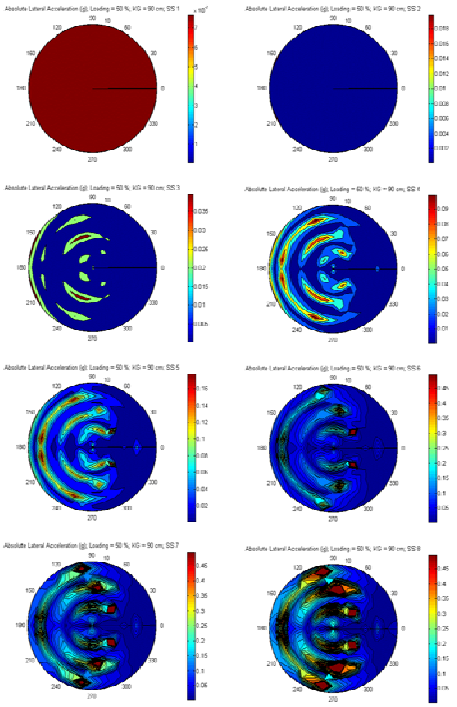


Figure 63. **Lateral acceleration at 50% midbox load.**

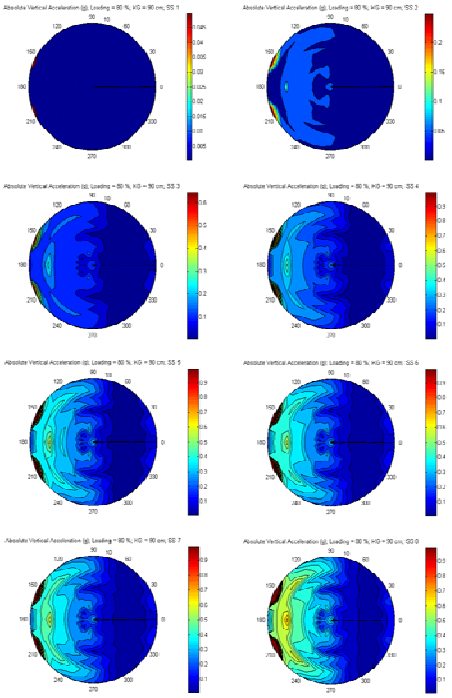


Figure 64. **Vertical acceleration at 80% midbox load.**

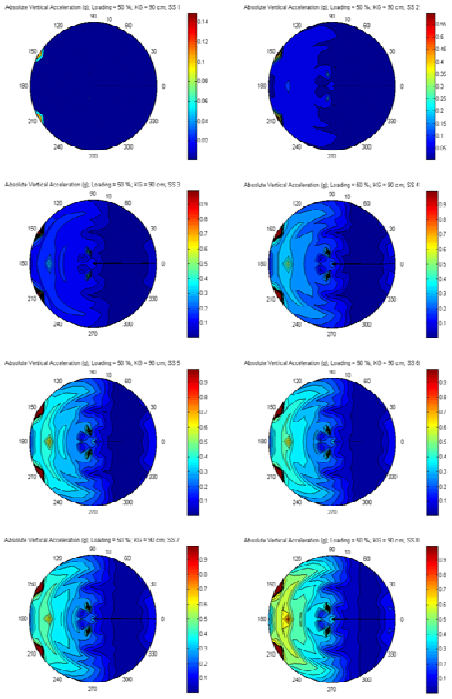


Figure 65. **Vertical acceleration at 50% midbox load.**

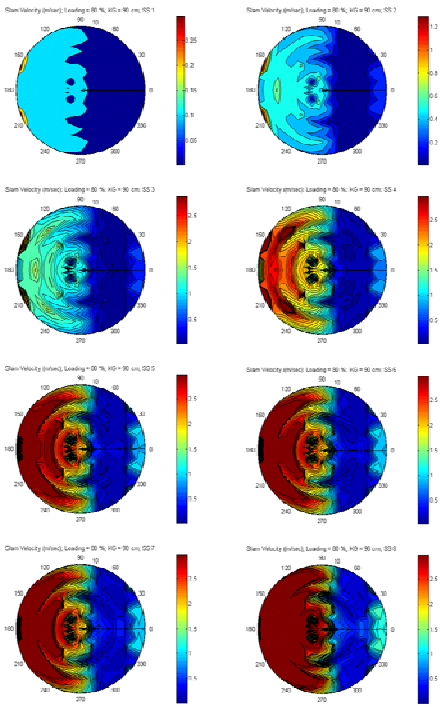


Figure 66. **Slam velocity at 80% midbox load.**

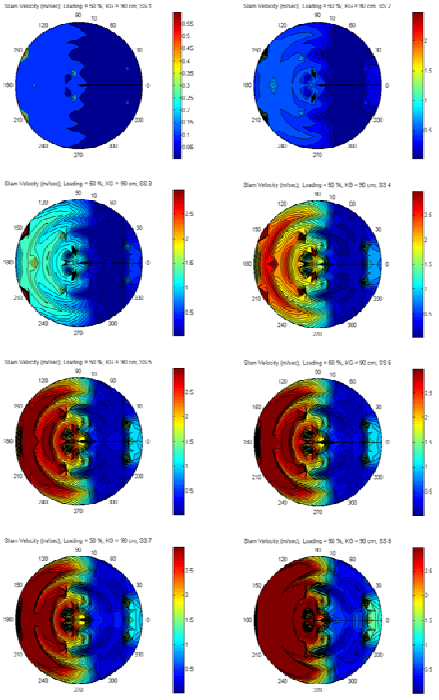


Figure 67. **Slam velocity at 50% midbox load.**

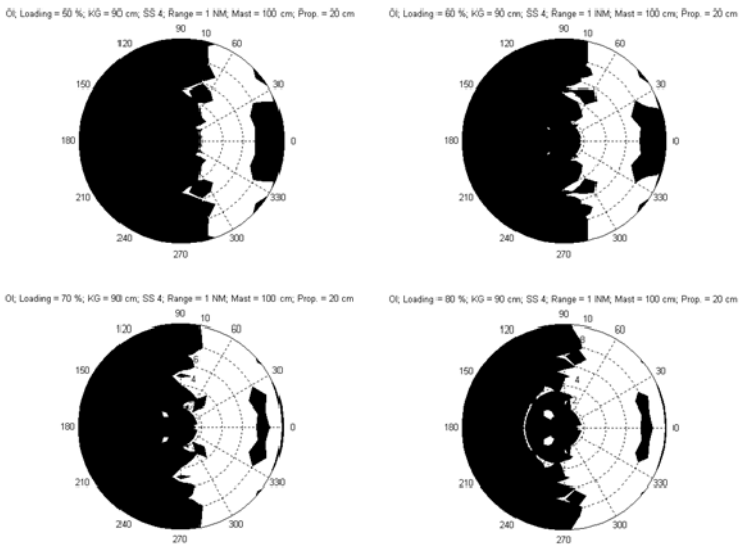


Figure 68. **Operability index for sea state 4, midbox load.**

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] E. V. Lewis. *Principles of Naval Architecture Volume II: Resistance, Propulsion, and Vibration.* The Society of Naval Architects and Marine Engineers, Jersey City, New Jersey. 1988.

[2] Robert F. Beck and Armin W. Troesch. *Documentation and User's Manual for the Computer Program SHIPMO.BM.* University of Michigan, Ann Arbor, September 1989.

[3] Garrett Mattingly. *The Defeat of the Spanish Armada.* Jonathan Cape, London. 1959.

[4] Federal Document. *Guadalcanal, The U.S. Army Campaigns of World War II.* U.S. Government Printing Office, Washington D.C. 2004.

[5] Ludwig Hough. *History of U.S. Marine Corps Operations in World War II: Pearl Harbor to Guadalcanal.* http://www.ibiblio.org/hyperwar/USMC/I/USMC-I-VI-9.html, 14 February 2008.

[6] Curtis A. Utz. *Assault from the Sea: The Amphibious Landing at Inchon.* Naval Historical Center, Washington D.C. 2000.

[7] Lynn Montross and Nicholas A. Canzona. *U.S. Marine Operations in Korea 1950-1953 Volume II: The Inchon-Seoul Operation.* Historical Branch, G-3 Headquarters U.S. Marine Corps, Washington D.C. 1955.

[8] Department of Homeland Security. *Navigation Rules, International-Inland COMDTINST M16672.2D.* Department of Homeland Security; United States Coast Guard, Washington D.C. 1983.

[9] Armin W. Troesch. *Naval Architecture 440, Marine Dynamics II.* University of Michigan, Ann Arbor. 1993.

[10] Naval Historical Center. *The Normandy Invasion, June 1944.* http://www.history.navy.mil, 21 February 2008.

[11] Royal Navy. *HMS Albion.* http://www.royalnavy,mod.uk, 21 February 2008.

[12] Mingtze Yeh. *Throughput Evaluation of an Autonomous Sustainment Cargo Container System.* Master's Thesis, Naval Postgraduate School. December 2007.

[13] Jesse Geisbert. *Autonomous Sustainment Cargo Container (ASCC) Resistance Evaluation.* NSWCCD: Code 5200 – Resistance and Powering. October 2007.

[14] W. M. Teppig, Jr. *Autonomous Movement of Containers from Ship to Shore.* Advanced Engineering & Planning (AEPCO) Inc., Gaithersburg, Maryland. July 2006.

[15] Caterpillar, Inc. *Marine Engine Application and Installation Guide – Ventilation Systems.* Peoria, Illinois. 2004.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Jeff Piatz
   Caterpillar Defense & Federal Products
   Peoria, Illinois

4. Fotis Papoulias
   Naval Postgraduate School
   Monterey, California